



PENERBIT ANDI



Shortcourse
Series

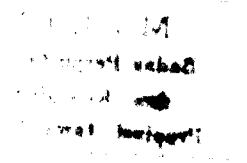


USTAKAAN
RSIPAN
WA TIMUR

74
P

PHP

Programming



PHP Programming

Diterbitkan Atas Kerjasama



WAHANA KOMPUTER



PENERBIT ANDI®



345.950/BPK/P/2010

ShortCourse: PHP Programming

Hak Cipta © 2009 pada WAHANA KOMPUTER.

Jl. MT. Haryono 637 Semarang Telp. (024) 8314727, 8413238, 8413963

Fax. (024) 8413964

Editor : C. Widyono Hermawan

Setting : Sri Sulistiyani

Desain Cover : Dany

Korektor : Marsi / Aktor Sadewa



Diterbitkan atas kerjasama Penerbit ANDI Yogyakarta dengan WAHANA KOMPUTER Semarang

Hak Cipta dilindungi undang-undang.

Dilarang memperbanyak atau memindahkan sebagian atau seluruh isi buku ini dalam bentuk apapun, baik secara elektronis maupun mekanis, termasuk memfotocopy, merekam atau dengan sistem penyimpanan lainnya, tanpa izin tertulis dari Penulis.

Penerbit: C.V ANDI OFFSET (Penerbit ANDI)

Jl. Beo 38-40, Telp. (0274) 561881 (Hunting), Fax. (0274) 588282
Yogyakarta 55281

Percetakan: ANDI OFFSET

Jl. Beo 38-40, Telp. (0274) 561881 (Hunting), Fax. (0274) 588282
Yogyakarta 55281

Perpustakaan Nasional: Katalog dalam Terbitan (KDT)

ShortCourse: PHP Programming /

— Ed. I. — Yogyakarta: ANDI; Semarang: WAHANA KOMPUTER;

18 17 16 15 14 13 12 11 10 09

xii + 250 + 10 hlm.; 16 x 23 Cm.

10 9 8 7 6 5 4 3 2 1

ISBN: 978 - 979 - 29 - 0869 - 5

I. Internet

1. WAHANA KOMPUTER Semarang

DDC'21: 004.678

Prakata

Sejalan dengan perkembangan teknologi web dan internet, berkembang pula bahasa pemrograman untuk membangun sebuah sistem informasi berbasis web. Cikal bakal bahasa pemrograman berawal dari munculnya HTML (HyperText Markup Language) -yang pada dasarnya bukan merupakan bahasa pemrograman, melainkan bahasa pengkodean. Perkembangan selanjutnya muncul bahasa pemrograman XML, Perl, Java Script, CGI Script, ASP, PHP, dan lain sebagainya. Pada umumnya, keamanan website database yang digunakan adalah: Microsoft Access, InterBase, mSQL, Sybase, Dbase, Informix, MySQL, SQL Server dan lain-lain.

Salah satu bahasa pemrograman yang saat ini terus berkembang dan paling banyak digunakan web programmer adalah PHP. PHP bersifat *open source* dan berada di bawah lisensi GPL (GNU Public License) sehingga Anda bebas menggunakan dan mendistribusikannya. Anda tidak perlu khawatir dengan proses instalasi pada sistem operasi karena mendukung semua varian Linux, Microsoft, Mac OS, dan varian UNIX. Selain Apache, jenis webserver yang dapat diaplikasikan adalah Personal Webserver, Netscape and Planet Servers, Xitami, OmniHTTP, dan Microsoft Internet Information server.

Buku *Short Course: PHP Programming* akan mengajak Anda belajar dan mendalami bahasa pemrograman PHP. Materi yang disajikan merupakan representasi dari bahan ajar yang dapat digunakan sebagai materi kursus yang meliputi: pengantar PHP, konfigurasi, konsep pemrograman PHP, hingga contoh aplikasi web dengan PHP. Setelah mempelajari buku ini, Anda diharapkan akan memperoleh pengetahuan dan penguasaan bahasa pemrograman PHP sebagai dasar untuk mengembangkan kemampuan programming web.

STRUKTUR PENULISAN BUKU

Buku *Short Course: PHP Programming* menjelaskan tentang dasar-dasar bahasa PHP dan segala kemampuan bahasa PHP yang berhubungan dengan perancangan web programming. Pembahasan materi disertai

dengan contoh kode program dan dibagi menjadi tujuh sesi yang semuanya dilengkapi dengan ilustrasi.

Sesi I Pengenalan PHP

Sesi ini merupakan bab perkenalan dengan teknologi PHP. Bab ini akan membahas sejarah perkembangan PHP, konsep PHP, fitur yang dimiliki, dan sebagainya.

Sesi II Konfigurasi PHP

Pada sesi ini, Anda akan mulai belajar mengenai konfigurasi PHP dan persiapan penggunaannya. Sajian materi yang diberikan adalah: kebutuhan sistem, aktivasi PHP 5, dan pengenalan XAMPP.

Sesi III Elemen dalam PHP

Sesi ini berisi beberapa elemen dan dasar pemrograman pada PHP, seperti: statement, expression, dan comment. Selain itu, bagian ini juga akan membahas tipe data dan variabel.

Sesi IV Konstanta, Operator, dan Ekspresi

Sesi ini berisi pembahasan mengenai konstanta, operator, ekspresi dan function. Pembahasan operator meliputi: operator aritmatika, logika, perbandingan, dan penugasan, sedangkan materi ekspresi yang dibahas adalah: increment, decrement, kondisional, perbandingan, hingga function.

Sesi V Struktur Kontrol PHP

Pada sesi ini, Anda akan belajar tentang struktur kontrol dalam PHP, seperti: konsep-konsep logika pemrograman dengan menggunakan struktur IF tunggal dan IF majemuk, switch case, serta contoh penerapan logika. Selain itu, bagian ini juga mengulas bentuk perulangan dengan menggunakan perulangan FOR, perulangan dengan WHILE, dan contoh penerapan Looping.

Sesi VI Function dan Class

Sesi ini akan membahas secara khusus function dan class dalam PHP, meliputi: konsep inheritance, polymorphism, constructor, dan destructor.

Sesi VII Aplikasi Web dengan PHP

Pada sesi ini, Anda akan mengenal contoh aplikasi web dengan PHP yang menggunakan konsep Multi Language Site, Captcha Security Images, dan Navigator Information.

APA YANG HARUS ANDA KUASAI?

Buku ini akan mengajarkan Anda pemrograman web dengan PHP. Penggunaan tag HTML pada pembahasan buku ini dilakukan untuk integrasi dengan PHP sebagai tampilan input dan proses data. Oleh karena itu, sebelum mempelajari buku ini Anda diharapkan telah mengenal konsep logika pemrograman dan dasar pemrograman HTML.

BAGI PARA PEMBACA

Walaupun dalam menyusun buku ini kami telah melakukan pengkajian dan penelitian yang mendalam serta berusaha untuk menyampaikan materi secara lengkap dan terstruktur, tetapi tentunya setiap karya tidaklah ada yang benar-benar sempurna, sehingga mungkin buku ini kurang dapat memenuhi kebutuhan para pembaca, atau mungkin para pembaca masih mengalami kesulitan atau masalah setelah mempelajari buku ini.

Untuk itu, Wahana Komputer membuka lebar-lebar kesempatan bagi para pembaca pada khususnya dan masyarakat pengguna komputer pada umumnya untuk melakukan konsultasi mengenai berbagai kesulitan yang dihadapi khususnya mengenai apa yang telah dijelaskan di dalam buku ini melalui situs web kami atau via pos, fax, ataupun e-mail. Di samping itu, Anda (para pembaca) dapat memanfaatkan pula layanan pelatihan komputer kami melalui Lembaga Pendidikan Komputer Wahana untuk lebih mendalami seluruh materi yang dipaparkan di dalam buku ini.

Wahana Komputer

JL. MT Haryono 637 Semarang

Tlp. (024) 8314727, 8413238 fax. (024) 8413964

e-mail : info@wahanakom.comweb : www.wahanakom.com

Daftar Isi

PRAKATA	iii
STRUKTUR PENULISAN BUKU	iii
APA YANG HARUS ANDA KUASAI?	v
BAGI PARA PEMBACA	v
DAFTAR ISI	vii
SESI I PENGENALAN PHP	1
1.1 PERKEMBANGAN PHP	3
1.2 SEJARAH PHP	4
1.2.1 <i>PHP/FI</i>	5
1.2.2 <i>PHP 3</i>	5
1.2.3 <i>PHP Versi 4</i>	5
1.2.4 <i>PHP Versi 5</i>	6
1.3 KONSEP PHP	7
1.4 PHP DENGAN BAHASA PEMROGRAMAN LAIN	8
1.4.1 <i>PHP dengan ASP</i>	8
1.4.2 <i>PHP dengan Cold Fusion</i>	9
1.4.3 <i>PHP dengan Perl</i>	9
1.5 KONSEP OPEN SOURCE	10
1.6 FITUR PHP	10
1.6.1 <i>Kelebihan PHP</i>	10
1.6.2 <i>Kelemahan PHP</i>	12
1.7 CARA KERJA PHP	14
1.7.1 <i>Membuat Web Dengan Skrip PHP</i>	16
1.8 RANGKUMAN	19
1.9 KUIS	20
1.10 LATIHAN	20
SESI II KONFIGURASI PHP	21
2.1 KEBUTUHAN PERANGKAT KERAS	22
2.2 KEBUTUHAN PERANGKAT LUNAK	22
2.3 MENGAKTIFKAN PHP 5	25
2.4 XAMPP	30

2.5	RANGKUMAN	32
2.6	KUIS	32
2.7	LATIHAN	32
SESI III	ELEMEN DALAM PHP	33
3.1	DASAR PEMROGRAMAN PHP.....	34
3.1.1	PHP dan Client Side Code	37
3.2	STATEMENT, EXPRESSION, DAN COMMENT	39
3.2.1	Statement	39
3.2.2	Expression	40
3.2.3	Comment	40
3.3	TIPE DATA	42
3.3.1	Tipe Data PHP	43
3.3.2	Konversi Tipe Data	67
3.4	VARIABEL	70
3.4.1	Konstanta.....	79
3.4.2	Mendefinisikan Konstanta.....	79
3.4.3	Variabel PHP.....	81
3.4.4	Variabel Melalui HTML (get dan post).....	81
3.5	RANGKUMAN	86
3.6	KUIS	87
3.7	LATIHAN	87
SESI IV	KONSTANTA, OPERATOR, EKSPRESI.....	89
4.1	KONSTANTA	90
4.1.1	Konstanta Predefined.....	96
4.2	OPERATOR.....	97
4.2.1	Operator Aritmatika	98
4.2.2	Operator Logika.....	102
4.2.3	Operator Perbandingan.....	105
4.2.4	Operator Penugasan.....	107
4.3	OPERATOR YANG LAIN.....	110
4.3.1	Unary Operator	111
4.3.2	String Concatenation Operator	112
4.3.3	Operator Bitwise.....	113
4.4	EKSPRESI.....	115
4.4.1	Ekspresi Increment dan Decrement	115
4.4.2	Ekspresi Kondisional.....	115
4.4.3	Ekspresi Perbandingan.....	116
4.4.4	Function	117

4.5	RANGKUMAN.....	118
4.6	KUIS	119
4.7	LATIHAN	119
SESI V	STRUKTUR KONTROL PHP	121
5.1	PERCABANGAN/BENTUK IF	123
5.1.1	Struktur If.....	123
5.1.2	Struktur If-Else.....	124
5.1.3	Struktur If-Elseif-Else.....	125
5.1.4	Struktur Switch...Case	134
5.2	LOOP/PERULANGAN.....	144
5.2.1	For	144
5.2.2	Foreach	159
5.2.3	While.....	162
5.2.4	do..while	166
5.2.5	Break.....	172
5.2.6	Continue.....	179
5.3	RANGKUMAN.....	185
5.4	KUIS	185
5.5	LATIHAN	185
SESI VI	FUNCTION DAN CLASS	187
6.1	FUNGSI	188
6.1.1	Fungsi Built-in	191
6.2	CLASS DALAM PHP	201
6.2.1	Class.....	202
6.2.2	Object.....	204
6.2.3	Properties.....	207
6.2.4	Inheritance.....	208
6.2.5	Polymorphism	210
6.2.6	Constructor	215
6.2.7	Destructor.....	217
6.3	RANGKUMAN.....	219
6.4	KUIS	219
6.5	LATIHAN	220
SESI VII	APLIKASI WEB DENGAN PHP	221
7.1	MULTI LANGUAGE SITE	222
7.2	CAPTCHA SECURITY IMAGES	225
7.3	NAVIGATOR INFORMATION.....	230

7.4 KALKULATOR DENGAN PHP	233
7.5 MEMBUAT KALENDER	236
7.6 ORGANISASI FILE.....	241
7.7 FILTER IMAGES DENGAN PHP.....	245
PENUTUP	249

Sesi I

Pengenalan PHP

Durasi

2 jam

Jenis

Teori dan Praktik

Media

PHP 5, Web Browser.

Tujuan

Setelah membaca sesi ini, siswa diharapkan dapat:

- ❖ Mengerti tentang sejarah PHP.
- ❖ Mengerti perkembangan PHP.
- ❖ Mengetahui kemampuan PHP.
- ❖ Mengetahui kelebihan dan kelemahan PHP.

Web merupakan media informasi berbasis jaringan komputer yang dapat diakses di mana saja dengan biaya relatif murah. Web merupakan bentuk implementasi dari bahasa pemrograman web (*web programming*). Sejarah perkembangan bahasa pemrograman web diawali dengan munculnya HTML (*Hypertext Markup Language*), yang kemudian dikembangkan dengan munculnya CSS (*Cascading Style Sheet*) yang bertujuan untuk memperindah tampilan website. Pada kenyataannya, bahasa tersebut masih memiliki kelemahan, yaitu skrip program dapat dilihat secara utuh, sehingga Anda dapat mengetahui kerangka situs tersebut dan memberi peluang hacker untuk mengubahnya dengan mudah.

Bahasa pemrograman saat ini sudah sangat berkembang dengan berbagai kemudahan dalam penyajian dan *interface* yang lebih *friendly used*. Penyajian yang baik dari suatu bahasa pemrograman akan menghasilkan sebuah web yang dinamis sehingga pengunjung akan lebih mudah mendapatkan informasi yang dibutuhkan.

Beberapa bahasa pemrograman web berbasis server (*server side*) mulai muncul dan dikembangkan oleh beberapa perusahaan perangkat lunak, seperti: ASP (*Active Server Pages*) oleh Microsoft, JSP (*Java Server Pages*) oleh Sun Microsystems, dan CGI (*Common Gateway Interface*).

Bahasa pemrograman tersebut memiliki beberapa kendala, misalnya: Active Server Pages keluaran Microsoft hanya dapat dijalankan melalui sistem operasi buatan Microsoft, dan Common Gateway Interface yang hanya berjalan pada sistem operasi berbasis UNIX. Untuk mengatasi keterbatasan tersebut, muncullah bahasa pemrograman web yang dapat dijalankan pada sistem operasi Microsoft, sekaligus sistem operasi berbasis UNIX, yaitu: bahasa pemrograman PHP (*PHP Hypertext Preprocessor*).

Skrip bahasa PHP adalah sama seperti skrip bahasa C. Bagi Anda yang sudah terbiasa menggunakan C, Anda dapat menggunakan PHP dengan mudah pula. Lebih dari itu, proses belajar PHP akan lebih mudah jika Anda telah menguasai bahasa pemrograman C++, Java, Pascal, atau Visual Basic, karena pada dasarnya logika pemrogramannya kurang lebih sama.

Tak perlu diragukan, PHP saat ini sudah sangat terkenal, bahkan bisa menyaingi popularitas bahasa pemrograman web lainnya dalam urusan web design. PHP kini lebih terkenal dari Perl dan ASP bahasa yang sebelumnya paling banyak dipakai untuk membuat aplikasi web dinamis.

Berdasarkan hasil survei Security Space (www.securityspace.com), PHP telah menjadi modul Apache paling populer yang digunakan para programmer, melebihi semua modul lain, seperti: Front Page atau Perl. Hal itu dibuktikan dengan seringnya php.net (situs homepage PHP) memajang hasil survei Netcraft (www.netcraft.co.uk) di halaman depan.

Banyak situs dirancang menggunakan PHP, mulai dari situs pribadi hingga situs dengan high security system, seperti: situs freephp.net, MTV Jerman, simtel.net, dan sourceforge.net. Portal Astaga! dan M-Web pun kini menggunakan PHP dan MySQL sebagai sistem basis data, berpindah dari software mahal seharga puluhan bahkan ratusan juta Vignette StoryServer dan Oracle yang sebelumnya dipakai situs astaga.com. Situs terkenal lain yang menggunakan PHP adalah naver.co.id, ekilat.com, dan liputan6.com. PHP cukup populer untuk forum atau situs komunitas, karena didukung dengan adanya aplikasi-aplikasi PHP, seperti: vBulletin, Phorum, atau phpBB yang dapat digunakan sebagai alternatif skrip Perl Ultimate Bulletin Board.

Banyak forum diskusi dan milis yang khusus dibuat sebagai wadah diskusi tentang PHP. Artikel mengenai PHP juga dapat dijumpai di berbagai media -baik online maupun offline. PHP telah tersedia di seluruh distro Linux/BSD, dan setiap hari instaler packing PHP untuk Windows pun di-download banyak penggunanya. Tak sulit mencari orang di lingkungan kampus/pendidikan, kelompok pengguna Linux, maupun di lapangan pekerjaan yang setidaknya mengenal atau pernah memakai PHP. Oleh karena itu, PHP banyak digunakan, baik dalam pemrograman web itu sendiri maupun di dunia pendidikan.

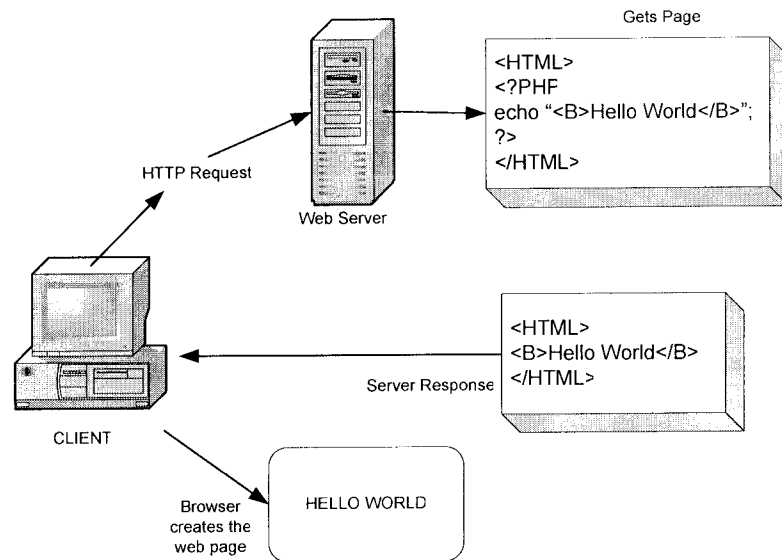
1.1 PERKEMBANGAN PHP

PHP (*Hypertext Preprocessor*) merupakan bahasa pemrograman berbasis web yang memiliki kemampuan untuk memproses dan mengolah data secara dinamis. PHP dapat dikatakan sebagai sebuah *server-side embedded script language*, artinya semua sintaks dan perintah program yang Anda tulis akan sepenuhnya dijalankan oleh server, tetapi dapat diletakkan pada halaman HTML biasa. Pada umumnya, semua aplikasi yang dibangun menggunakan PHP akan memberikan hasil pada web browser, tetapi prosesnya secara keseluruhan dijalankan pada server.

Pada prinsipnya, server akan bekerja apabila ada permintaan dari client. Dalam hal ini, client menggunakan kode-kode PHP akan mengirim permintaan ke server (perhatikan Gambar 1.1). Ketika menggunakan

PHP sebagai *server-side embedded script language*, maka server akan melakukan beberapa hal sebagai berikut:

- Membaca permintaan dengan skrip PHP berasal dari browser.
- Mencari halaman/page di server (*server pages*).
- Melakukan *processing* melalui instruksi yang diberikan oleh PHP untuk melakukan modifikasi pada halaman/page.
- Mengirim kembali halaman tersebut kepada client melalui internet atau intranet yang merupakan proses echo/print.



Gambar 1.1 Siklus PHP dalam Web Server

1.2 SEJARAH PHP

PHP memiliki sejarah yang cukup panjang, berawal dari gagasan seorang programmer Bahasa C bernama Rasmus Lerdorf. Dalam perjalanannya, banyak orang yang mengajak kerja sama sehingga PHP semakin cepat berkembang dan mampu mengalahkan pesaingnya.

1.2.1 PHP/FI

Bermula sekitar tahun 1994, Rasmus Lerdorf menggunakan PHP untuk menghitung jumlah pengunjung di dalam situsnya. Kemudian ia mengeluarkan Personal Home Page Tools versi 1.0 secara gratis. Versi ini pertama kali keluar pada tahun 1995 dan berisi sekumpulan skrip PERL yang digunakan untuk membuat halaman situsnya menjadi dinamis. Selanjutnya, pada tahun 1996 ia mengeluarkan PHP versi 2.0 yang telah memiliki kemampuan untuk mengakses database dan dapat terintegrasi dengan HTML.

Selanjutnya, dirilis paket tool **Personal Home Page**, yang merupakan cikal bakal lahirnya PHP. Rasmus Lerdorf kemudian menulis *engine scripting* dan menggabungkan tool yang berbeda untuk *input parse* dari HTML, yang biasa disebut dengan FI (*Form Interpreter*). Pada akhirnya, versi ini disebut dengan **PHP/FI** atau **PHP2** yang dirilis pada sekitar tahun 1995.

1.2.2 PHP 3

Pada tahun 1997 muncul PHP/FI versi 2.0, dengan seluruh sintaks dan fungsi yang ditulis menggunakan bahasa C. Karena domain yang menggunakan PHP/FI sangat besar, maka berkat bantuan dari berbagai pihak yang beranggotakan Andi Gutmans, Zeev Suarasski, Shane Caraveo, Stig Bakken, dan Jim Winstead, kemudian mengawali lahirnya PHP 3. PHP 3 terdapat penambahan API, fungsi dan modul dalam script engine PHP yang turut berpartisipasi mengembangkan PHP/FI. Pada tanggal 6 Juni 1998 muncul **PHP versi 3.0** yang dikeluarkan oleh Rasmus bernama kelompok pengembang software-nya.

1.2.3 PHP VERSI 4

Versi berikutnya adalah **PHP 4.0**, yang dirilis pada tanggal 22 Mei 2000 dan merupakan versi yang lebih lengkap dibandingkan versi sebelumnya. Perubahan yang paling mendasar pada PHP 4.0 adalah terintegrasinya *Zend Engine* yang dibuat oleh Zend Suraski dan Andi Gutmans, yang merupakan penyempurnaan dari PHP3 scripting engine. Fitur baru lainnya adalah *build in HTTP session*, dan tidak lagi menggunakan *library* tambahan seperti pada PHP3. Tujuan dari bahasa scripting ini adalah untuk membuat aplikasi-aplikasi yang dijalankan pada teknologi web.

Aplikasi pada umumnya akan memberikan hasil pada web browser, tetapi prosesnya secara keseluruhan dijalankan web server.

PHP 4 diluncurkan dalam rangka mengatasi kelemahan-kelemahan yang terdapat pada PHP 3, yaitu dengan penambahan fitur baru, seperti: session, output buffering, dan dalam penanganan input output. PHP 4 juga disediakan dalam paket yang dibuat oleh seorang programmer bernama J Wynn berkebangsaan Czech (Czech Republic Slovakia), yakni paket PHPTriad.

1.2.4 PHP VERSI 5

PHP 5 adalah versi yang dirilis pada awal tahun 2006 dan pada bulan Desember 2008 telah muncul hingga versi 5.2.8. Dengan berbagai kelebihan dibandingkan versi sebelumnya, PHP 5 hadir membawa lebih banyak fungsi dan perbaikan terhadap bug. Dengan PHP 5, Anda akan mampu menangani dan membuat session pada objek serta model keamanan menggunakan md5 dan seterusnya. Selain itu, ada juga tambahan library, seperti: GD2 untuk memproses *dynamic image* dengan lebih efektif. Fungsi untuk menangani Form dan Simple XML juga ditingkatkan pada versi 5 dan masih banyak kelebihan lainnya dan tambahan fungsi yang dapat mempermudah untuk bekerja dengan PHP & MySQL. PHP 5 lebih cepat, stabil, dan siap untuk digunakan untuk skala produksi.

Beberapa peningkatan fungsionalitas dan penambahan beberapa fitur pada PHP 5 adalah:

- Peningkatan kemampuan pemrograman berorientasi objek (OOP).
- Exception handling, sistem standarisasi logika atas pemberitahuan kesalahan pemrograman yang belum didukung oleh PHP 4.
- Peningkatan terhadap penanganan *string* lebih lanjut.
- Peningkatan dukungan terhadap XML dan Web Service, yang menggunakan libxml2.
- Dukungan terhadap SQLite database server.

1.3 KONSEP PHP

Rancangan dan konsep bahasa PHP sangat sederhana, bahkan lebih sederhana daripada CGI. Cukup membuat sebuah HTML biasa, dengan menambahkan kode-kode program yang diapit dalam tanda `<? ... ?>`, maka jadilah skrip PHP. Persis seperti Javascript, hanya berbeda pengapit dan dieksekusi di sisi server, sehingga saat tiba ke browser kode PHP sudah diganti dengan hasil eksekusinya. Sementara itu, untuk membuat aplikasi CGI, Anda harus mengerti dulu mekanisme HTTP dan protokolnya, apa itu variabel environment, atau bagaimana mencetak header respon HTTP yang benar, sehingga cukup merepotkan.

Untuk menjadi seorang programmer, khususnya pemrogram web, semua teori tersebut harus dipahami, namun dengan PHP tidak diwajibkan dari awal. Hal yang cukup menggembirakan bagi seorang pemula, tidak akan lagi muncul halaman notifikasi error bertuliskan: "500 Internal Server Error". Seperti pada pemrograman CGI tradisional, halaman yang dingin, tidak ramah, tanpa dilengkapi keterangan tambahan apa-apa tersebut muncul setiap kali terjadi kesalahan sintaks atau terlupanya header yang benar dicetak. Seorang programmer harus masuk ke server melalui telnet/SSH untuk melihat pesan kesalahannya di file log webserver. PHP membantu pemula dengan perlakuan default, yaitu mencetak pesan kesalahan ke browser. Ini akan mempercepat siklus *debugging*. Dengan demikian, kehadiran PHP yang terbaru saat ini telah memudahkan programmer web -yang sebelumnya sudah antipati dengan Perl/CGI- dalam pembuatan skrip web, setidaknya untuk membuat halaman minimal dinamik yang sederhana, seperti: sistem include header-footer, form mailer, guestbook, atau news headlines grabber. Dengan menambahkan fungsi fopen() di PHP misalnya, langsung dapat membuka dokumen URL remote.



Gambar 1.2 Logo PHP

Pada dasarnya, skrip PHP merupakan saingan berat skrip ASP. Padahal, kedua bahasa pemrograman tersebut memiliki kesamaan, yaitu dapat diterapkan pada skrip HTML dan dijalankan oleh webserver. Sebagai saingan berat ASP, PHP memiliki beberapa keunggulan, di antaranya:

- Relatif Aman, mengingat skrip atau source code PHP yang terdapat pada web browser tidak dapat dilihat dengan fasilitas View HTML source, seperti pada Internet Explorer atau sejenisnya karena skrip PHP ditempelkan pada tag HTML.
- Fleksibel, instalasi sistem operasi mendukung semua varian Linux, Windows, Mac OS, dan varian UNIX. Selain itu, jenis webserver yang dapat diaplikasikan selain Apache adalah Personal Webserver, Netscape and Planet Servers, Xitami, OmniHTTP, dan Microsoft Internet Information server.
- Dalam hal keamanan data, PHP dapat menggunakan berbagai macam varian database, seperti: MySQL, Microsoft Access, InterBase, mSQL, Sybase, Dbase, Informix, SQL Server, dan lain-lain.
- PHP dapat berintegrasi dengan semua aplikasi program CGI, misalnya mengambil nilai form, sehingga menghasilkan halaman web yang dinamis. Selain itu, integrasi dapat juga dengan mengirimkan dan menerima cookie. PHP juga dapat berkomunikasi dengan berbagai layanan yang menggunakan protokol HTTP, SNMP, NNTP, POP3, IMAP, dan lain sebagainya.
- PHP bisa dikatakan cepat karena ditempelkan pada HTML sehingga waktu tanggap menjadi pendek.
- Dilihat dari segi harga PHP tidak mahal, malah kebanyakan varian PHP gratis dan dapat diperoleh dengan cuma-cuma.
- Selain bisa didapatkan secara gratis karena di bawah lisensi GNU, Anda juga dapat memodifikasi fitur PHP sesuai dengan kebutuhan.

1.4 PHP DENGAN BAHASA PEMROGRAMAN LAIN

Sebagai salah satu bahasa pemrograman - khususnya dalam lingkup web programming- PHP tentu memiliki pesaing, yang dalam perkembangannya selalu dituntut untuk berkompetisi.

1.4.1 PHP DENGAN ASP

ASP (Active Server Pages) tidaklah benar-benar suatu bahasa yang dapat berdiri sendiri, yang menggunakan Visual Basic dan JScript dalam pemrogramannya. Kelemahan terbesarnya, ASP hanya dapat dijalankan

melalui Microsoft Internet Server Information (IIS). Ada beberapa proyek yang mengizinkan ASP untuk singgah ke lingkungan dan webserver yang lain. ASP juga disebut sebagai bahasa pemrograman yang lebih sulit dan lebih lambat dibanding PHP, juga lebih sedikit yang stabil. ASP juga sangat mungkin dijalankan di IIS server sehingga membuatnya mudah untuk dibangun dan dijalankan. Komponen yang dibangun di dalam ASP sangat dibatasi. Jadi, jika Anda menggunakannya untuk tingkat mahir seperti penggunaan interface dengan FTP Server, Anda harus membeli komponen tambahan.

1.4.2 PHP DENGAN COLD FUSION

Secara umum, PHP dikenal sebagai bahasa yang lebih stabil dan lebih sedikit program tambahan yang dibutuhkan. Kekuatan Cold Fusion terletak pada kesempurnaan Search Engine dan penanganan kesalahan/Error Handling. Namun demikian, penggunaan *search engine* bukan menjadi suatu hal yang sangat penting dalam pemrograman web, bahkan dalam perancangan database dan struktur penanggalan justru lebih banyak diuraikan pada PHP. PHP dapat berjalan hampir di setiap platform, sedangkan Cold Fusion hanya tersedia pada sistem dengan format Win32, seperti: Solaris, Linux, dan HP/UX. Cold Fusion mempunyai suatu IDE yang baik dan sangat mudah dipelajari, sedangkan PHP pada awalnya memerlukan pengetahuan yang lebih dalam bahasa pemrograman. Cold Fusion didesain bukan berdasarkan pemikiran seorang programmer, sedangkan PHP fokus pada programmer, khususnya pemrograman web.

1.4.3 PHP DENGAN PERL

PHP dan Perl merupakan bahasa pemrograman yang serupa karena keduanya berasal dari bahasa C. PHP juga mengambil sedikit prefix \$ untuk penamaan variabel, gaya, dan beberapa fungsi pengolahan string. Keuntungan yang paling besar tentang PHP yang melebihi Perl adalah bahwa PHP telah dirancang dengan skrip yang dapat dipadupadankan dengan Perl. Fleksibilitas/kompleksitas Perl membuatnya lebih mudah untuk dituliskan berdasarkan pembuat bahasa pemrograman, sehingga menyulitkan programmer untuk menulis skrip pada Perl. PHP mempunyai lebih ringan namun tanpa kehilangan fleksibilitas. PHP lebih mudah untuk diintegrasikan ke dalam HTML yang ada dibanding Perl. PHP

mempunyai banyak kemampuan luar biasa dibanding Perl dalam penulisan sintaks dan logika pemrograman. Pada akhir tahun 1980-an, Perl banyak digunakan, namun setelah kemunculan PHP, Perl sudah banyak ditinggalkan.

1.5 KONSEP OPEN SOURCE

GNU GPL (GNU General Public License) adalah lisensi bebas atau free software license, yang pada mulanya dicetuskan dan digunakan oleh Richard Stallman untuk melindungi proyek GNU-nya. Sejak saat itu, kemunculan GNU sangat populer sebagai acuan untuk perlindungan free software (*freeware*).

Versi 2 dirilis tahun 1991 dan merupakan versi terakhir. Versi lain adalah GNU Lesser General Public License (LGPL), yang sering diterapkan dalam perlindungan sejumlah software libraries.

Pada prinsipnya, GPL menjamin pengguna komputer mendapatkan beberapa hal, yakni:

- Kebebasan untuk menjalankan program pada berbagai keperluan.
- Kebebasan untuk mempelajari cara kerja program tersebut, memperbolehkan modifikasi dan mewajibkan memberi akses untuk kode sumbernya kepada publik.
- Kebebasan untuk me-redistribusi-kan salinannya.
- Kebebasan untuk memperbaiki, meningkatkan, dan merilis perbaikannya untuk dipublikasikan.

1.6 FITUR PHP

Sebagai bahasa pemrograman tingkat tinggi, PHP tentu memiliki nilai lebih, khususnya dalam skrip dan tingkat kesulitan serta fleksibilitas bahasa tersebut ketika dimodifikasi.

1.6.1 KELEBIHAN PHP

PHP (Hypertext Processor) mempunyai beberapa kemampuan yang merupakan salah satu kelebihan PHP. Kemampuan tersebut antara lain:

- Cara koneksi dan *query* database yang sederhana.

- Dapat bekerja pada sistem operasi berbasis Windows, Linux, Mac OS, dan kebanyakan varian UNIX.
- Biaya yang dibutuhkan untuk menggunakan PHP tidak mahal, atau bahkan gratis.
- Mudah digunakan karena memiliki fitur dan fungsi khusus untuk membuat web dinamis. Bahasa pemrograman PHP dirancang untuk dapat dimasukkan dalam HTML (*embedded script*).
- Security system yang cukup tinggi.
- Waktu eksekusi yang lebih cepat dibandingkan dengan bahasa pemrograman web lainnya berorientasi pada *server side scripting*.
- Akses ke sistem database yang lebih fleksibel dan mudah, seperti pada MySQL.

Ketika kebutuhan informasi yang *up-to-date* dan kegiatan e-commerce semakin berkembang, membuat situs-situs statis pun semakin ditinggalkan karena dianggap sudah tidak memenuhi kebutuhan pasar. Saat ini, bahasa PERL dan CGI sudah ketinggalan jaman sehingga sebagian besar web programmer banyak yang beralih ke bahasa *server side scripting* yang lebih dinamis seperti PHP.

Seluruh aplikasi berbasis web dapat dibuat menggunakan PHP. Kelebihan utama PHP terletak pada kemudahan koneksinya dengan berbagai sistem database di dalam web. Sistem database yang dapat didukung oleh PHP adalah:

- Adabas D
- Direct MS-SQL
- Oracle (OCI7 and OCI8)
- dBase
- InterBase
- Informix
- ODBC
- Hyperwave
- Sybase
- IBM DB2
- MySQL
- FrontBase
- PostgreSQL

- mSQL Solid
- dll.

Selain dukungan koneksi sistem database yang cukup beragam, Anda juga dapat menggunakan PHP untuk berhubungan dengan berbagai macam protokol, seperti: IMAP, NNTP, LDAP, HTTP, SNMP, COM, dan lainnya. Fasilitas lainnya adalah socket programming, yang dapat digunakan untuk membuat skrip php yang dapat menghubungkan komputer pada sebuah jaringan, sehingga Anda bisa bertukar data (data sharing).

PHP memiliki banyak tipe function yang memiliki kegunaan:

- DOM XML Functions.
- NET Functions.
- Filesystem Functions.
- COM (untuk Windows), untuk berkomunikasi atau mengirim data antaraplikasi yang dibuat dengan standar com.
- ClibPDF Functions, yang digunakan untuk membuat file pdf secara *on the fly*. Fungsi ini mirip dengan PDFlib.
- Cybercash Payment Functions, yang digunakan programmer web untuk membuat web yang berhubungan dengan E-Payment.
- Bzip2 Compression Functions, fungsi yang digunakan PHP untuk membuat dan membaca file berformat *.zip.

PHP merupakan software *open source* yang dapat di-download secara gratis dari situs resminya, yaitu: <http://www.php.net>. Software ini juga dapat berjalan pada web server, seperti: PWS (Personal Web Server), Apache, IIS, AOLServer, fhttpd, phttpd, dan sebagainya. PHP juga merupakan bahasa pemrograman yang dapat dikembangkan, misalnya dengan penambahan fungsi-fungsi baru. Keunggulan lainnya adalah: PHP juga mendukung komunikasi dengan layanan, seperti: protocol IMAP, SNMP, NNTP, POP3, bahkan HTTP. PHP dapat diinstal sebagai bagian atau modul dari Apache Web Server atau sebagai CGI skrip yang mandiri.

1.6.2 KELEMAHAN PHP

Selain memiliki berbagai keunggulan, PHP juga memiliki beberapa kekurangan. Dari segi bahasa, PHP bukanlah bahasa yang cocok untuk pengembangan berskala besar. Kekurangan yang utama adalah tidak adanya namespace. **Namespace** adalah sebuah cara untuk mengelompokkan fungsi atau nama variabel dalam susunan hierarki. Sebagai

contoh, `$ScriptA::ScriptB::c` adalah variabel `$c` di dalam `ScriptB`, yang terdapat di bawah `ScriptA`. Script lain, misalnya `ScriptA::ScriptC` atau `ScriptD` dapat memiliki variabel `$c` juga, tetapi keduanya merupakan dua variabel berbeda. Di dalam `ScriptB`, variabel `$c` pertama dapat disebutkan hanya `$c`, tetapi untuk merujuk pada variabel kedua, diperlukan nama lengkap `$ScriptD::c`. Di PHP ini tidak ada, jadi kadang programmer menyiasatinya dengan membuat nama seperti `$ScriptA_ScriptB_c`. Kekurangannya, nama yang panjang ini harus disebutkan di setiap waktu, namun sekarang PHP 5 sudah memiliki solusi, yaitu dengan menggunakan `include`.

Pada PHP, Anda tidak dapat membuat fungsi di dalam fungsi atau kelas di dalam kelas, semuanya hanya terbatas satu level. Sebagian programmer tingkat lanjut mengeluhkan model objek PHP yang minim. Namun dengan munculnya PHP 5, semua permasalahan mengenai keterbatasan objek sudah dapat terjawab. Dibandingkan dengan Python dan Perl, PHP tidak memiliki *multiple inheritance* atau kemampuan untuk mewarisi dua atau lebih kelas induk. Multiple inheritance dapat berguna untuk fleksibilitas dalam pengembangan. Java memang tidak memiliki multiple inheritance, tetapi ada interface dalam Java. Di PHP ini pun tidak ada.

Selain itu, belum ada penanganan eksepsi (raise, catch, throw) di PHP. Anda tidak bisa memanggil metode/atribut secara berangkai, misalnya: `$obj->meth()->meth2()`. Hal lainnya, tidak ada destruktur (meskipun ada konstruktor) yang terdapat pada PHP 4, sedangkan dalam PHP 5 sudah terdapat destruktur. Pengembang harus berhati-hati agar aplikasi yang dikembangkan tidak beraturan, mengingat model pemrograman yang *embedded* (tercampur dengan HTML). Kombinasi dengan HTML yang rumit tersebut dapat mengurangi modularitas, reuse, dan membebani maintenance. Oleh karenanya, upaya pemisahan kembali antara template HTML dan kode/logika yang terdapat pada PHP perlu dilakukan untuk aplikasi berskala menengah ke atas, antara lain dengan sistem template.

Sebagai konsekuensi atas kepraktisan dan kemudahannya, instalasi *default* pada PHP banyak memiliki kelemahan keamanan (security system). Variabel global di PHP dapat berasal dari input pengunjung Web (dari GET/POST/Cookie), sehingga bila seorang programmer ceroboh tidak menginisialisasi tiap variabel sebelum pemakaian, seseorang dapat menyerang dengan memasukkan nilai-nilai awal variabel ke dalam skrip untuk mengubah kelakuannya.

Sebelum PHP 3.0.18, terdapat bug pada file upload yang banyak dieksploitasi untuk menembus banyak situs PHP. Dalam bug ini, interpreter PHP dapat diakali untuk menulis file pada file system server manapun, sesuai keinginan penyerangnya, karena path dapat dimasukkan lewat form HTML. Beberapa kelemahan tersebut dapat dikonfigurasi atau dimatikan. Oleh karenanya, seorang programmer PHP dan admin perlu mengetahui opsi-opsi konfigurasi PHP agar sistem mereka lebih aman.

Dibandingkan dengan aplikasi CGI dalam Perl atau Python -apalagi yang berukuran cukup besar, aplikasi serupa dalam PHP sedikit banyak berjalan lebih cepat. Pada umumnya, PHP dipasang sebagai sebuah modul Apache, sehingga webserver dapat langsung mengerti dan memroses skrip PHP karena interpreter PHP telah ditanamkan ke dalamnya. Hal ini dilakukan untuk menghindari *bottle neck* yang pada umumnya terjadi di Unix, yaitu proses *forking interpreter* di setiap *request*. Pada dasarnya, PHP juga dapat dijalankan melalui mekanisme CGI, tetapi bahasa PHP yang lebih sederhana menyebabkan skrip PHP dapat diinterpretasikan dalam waktu lebih singkat.

Selain kemampuan tersebut, PHP juga memiliki kekurangan pada saat instalasi dan konfigurasi yang cukup rumit dan berbeda dengan sistem operasi yang digunakannya karena dapat bekerja pada *multi-platform*.

1.7 CARA KERJA PHP

Untuk menciptakan situs web yang dinamis dengan PHP, Anda harus mengetahui bagaimana cara mengirimkan data kepada web browser. PHP mempunyai sejumlah *built-in function* untuk tujuan pengiriman data ini, yang paling umum menggunakan *echo()* dan *print()*; kebanyakan programmer lebih menyukai *echo()*:

```
Echo 'Selamat belajar PHP 5!';
```

```
Echo "Apa yang baru di PHP 5";
```

Anda bisa menggunakan *print()*, seperti:

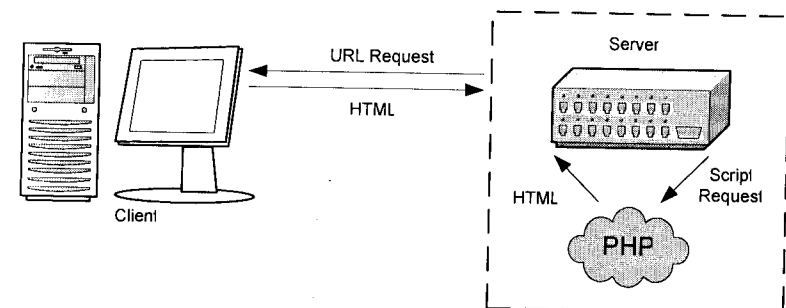
```
Print "Selamat belajar PHP 5";
```

```
print "Apa yang baru di PHP5";
```

Pada contoh tersebut, Anda dapat menggunakan tanda kutip tunggal atau ganda. Tanda kutip pertama setelah nama fungsi menandai adanya

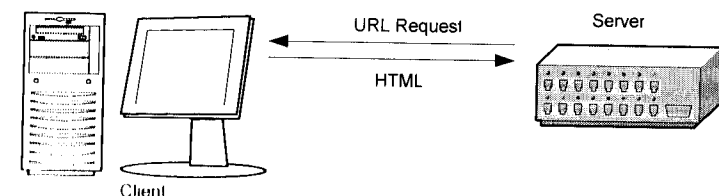
pesan yang akan dicetak, sedangkan mempertemukan tanda kutip yang menandakan bahwa skrip di antara tanda kutip yang akan dicetak. Bersama dengan cara mengirimkan data pada web browser, Anda perlu mencatat juga bahwa semua statement harus diakhiri dengan sebuah titik koma/semicolon. Selain itu, PHP bersifat case-insensitive saat digunakan untuk mengeksekusi suatu fungsi.

Sebagai bahasa *server-side*, kode yang Anda buat dalam PHP diakses oleh user melalui server, kemudian server mengirim permintaan halaman web yang dikunjungi. Ketika user mengunjungi halaman web yang ditulis dalam PHP, server membaca skrip PHP, lalu memrosesnya menurut arah skripnya. Pada **Gambar 1.3**, skrip PHP memanggil server untuk kemudian mengirim data dalam bentuk skrip HTML agar dapat dijalankan dalam web browser.



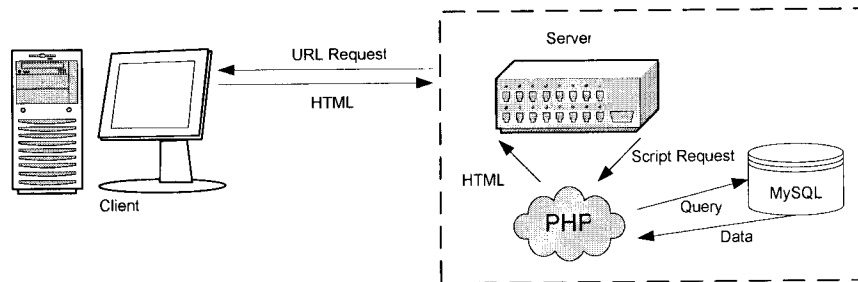
Gambar 1.3 PHP Terhubung dengan Client/Server saat Mengakses Halaman Web

Berbeda halnya dengan HTML, yang akan berbentuk statis untuk membangun suatu web. Ketika suatu permintaan dari client dibuat kepada server, server mengirim skrip HTML ke dalam web browser dan di sana tidak ada konversi skrip untuk menjadi HTML, sehingga halaman yang muncul dalam browser sama persis seperti yang terdapat dalam skrip.



Gambar 1.4 Client/Server meminta pada Halaman Web Statis yang dibuat dalam HTML

Jika terdapat database pada situs yang dibuat, maka pada saat client melakukan permintaan kepada server, server akan memproses apakah pada permintaan tersebut terdapat aktivitas input-output data. Selanjutnya, data tersebut akan disimpan dalam database. Dalam hal ini, Anda menggunakan database MySQL, sehingga Anda telah menciptakan suatu web yang dinamis.



Gambar 1.5 Siklus Kerja Web Dinamis dengan PHP dan MySQL

1.7.1 MEMBUAT WEB DENGAN SKRIP PHP

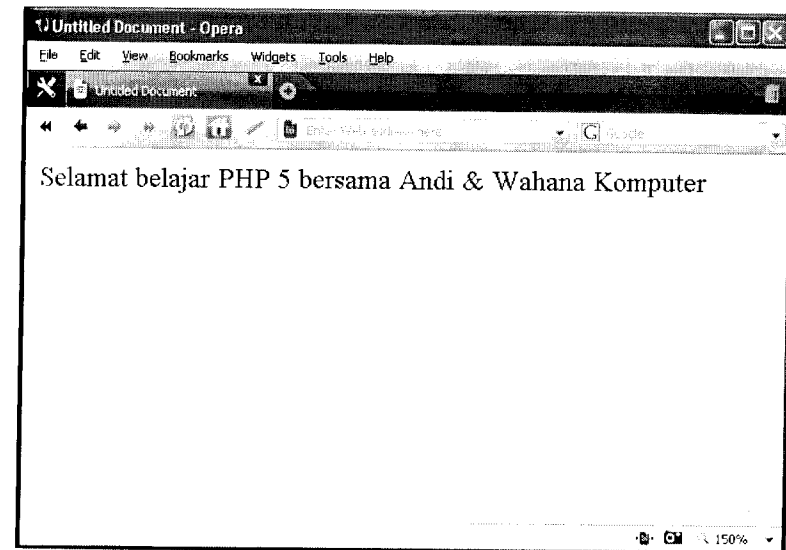
Bagaimana menuliskan sintaks PHP pada HTML? Langkah awal yang dapat dilakukan adalah membuat sintaks dengan tag `<?;`, kemudian diakhiri dengan tag `?>`. Aturan penulisan yang akan ditampilkan dalam server dapat menggunakan `print` atau `echo`, seperti pada contoh skrip program PHP berikut:

```
<html>
  <head>
    <title>HTML dan PHP terintegrasi</title>
  </head>
  <body>
    <?php
      echo "Selamat belajar PHP 5 bersama Andi & Wahana
Komputer" ;
    ?>
  </body>
</html>
```

Contoh program :

```
<html>
  <head>
    <title>HTML dan PHP terintegrasi</title>
  </head>
  <body>
    <?php
      print "Selamat belajar PHP 5 bersama Andi & Wahana
Komputer" ;
    ?>
  </body>
</html>
```

Selanjutnya, Anda akan melihat tampilan seperti yang terlihat pada Gambar 1.6 saat dijalankan pada browser.



Gambar 1.6 Tampilan PHP Sederhana

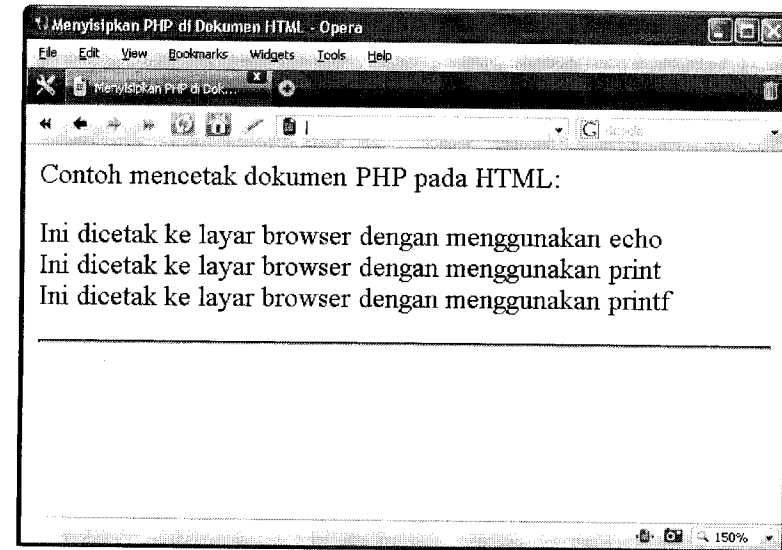
Contoh skrip PHP tersebut akan menghasilkan output yang sama pada web browser. Dengan demikian, penggunaan *echo()* dan *print()* adalah sama, mengingat keduanya akan mencetak halaman yang diapit dengan tanda kutip.



Pencarian error dalam penulisan skrip PHP biasanya terdapat pada baris yang terdapat error atau baris sebelumnya.

Contoh program:

```
<HTML>
<HEAD>
<TITLE>Menyisipkan PHP di Dokumen HTML</TITLE>
</HEAD>
<BODY>
Contoh mencetak dokumen PHP pada HTML: <BR>
<P>
<?php
echo("Ini dicetak ke layar browser dengan menggunakan echo");
echo("<BR>");
print("Ini dicetak ke layar browser dengan menggunakan print");
print("<BR>");
printf("Ini dicetak ke layar browser dengan menggunakan printf");
printf("<HR>");
?>
</BODY>
</HTML>
```



Gambar 1.7 Hasil Cetak dengan echo, print, printf

Dari contoh penggunaan *echo("
")*, *print("
")*, dan *printf("
")*, ketiganya mengeksekusi tag HTML, yaitu tag **
** yang berfungsi untuk mengganti baris. Ini menunjukkan fungsi dari ketiganya dalam menjalankan tag HTML di PHP.



Secara teknis, *echo()* dan *print()* adalah skrip yang digunakan untuk membangun, bukan sebagai fungsi, sehingga dalam penggunaannya dapat sesuai dengan kebutuhan. *Echo()* dan *print()* dapat digunakan untuk mencetak teks lebih dari 1 baris. *echo* "kalimat ini akan dicetak dalam dua baris."; *print* "kalimat ini akan dicetak dalam dua baris.";

1.8 RANGKUMAN

Teknologi PHP yang semakin berkembang membuat bahasa pemrograman ini banyak digunakan oleh domain. Dilihat dari segi biaya, bisa dikatakan sangat murah -bahkan gratis. Dalam hal keamanan, kemampuan PHP dapat diterapkan pada berbagai jenis database, seperti: Microsoft Access, InterBase, mSQL, Sybase, Dbase, Informix, MySQL, SQL Server, dan lain-lain. Dalam penulisan skrip, jika Anda terbiasa menggunakan Pascal, C, dan lain sebagainya akan lebih mudah karena

logika pemrogramannya hampir mirip. Dengan demikian, maka keputusan untuk merancang suatu website dengan pendukung bahasa PHP dan menggunakan database yang menurut Anda paling aman dan paling cocok untuk keamanan sangat tepat. Selain aman dari segi keamanan data, skrip PHP juga tidak dapat dilihat dari web browser, seperti halnya pada Internet Explorer karena yang muncul adalah tag HTML.

Beberapa kelemahan yang disebutkan sebelumnya tentu saja tidak menghalangi PHP untuk tetap digunakan secara meluas. Setidaknya, dibandingkan dengan produk-produk komersial, PHP dapat menurunkan biaya karena software-nya gratis dan sudah mencakup fitur yang cukup lengkap.

1.9 KUIS

1. Jelaskan sejarah perkembangan PHP!
2. Mengapa banyak domain yang menggunakan PHP?
3. Sebutkan kemampuan yang dimiliki oleh PHP!
4. Apakah dokumen PHP dapat dibuat tanpa menggunakan tag HTML? Jelaskan!
5. Sebutkan keunggulan PHP dari bahasa pemrograman web yang lain!

1.10 LATIHAN

Buatlah skrip PHP yang menampilkan hasil pada browser, dengan ketentuan:

Nama : ...(Diisikan nama Anda)...

TTL : ...(Diisikan tempat tanggal lahir Anda)...

Alamat : ...(Diisikan alamat Anda)...

Email : ...(Diisikan email Anda)...

Gunakan *echo*, *print*, dan *printf*, simpan dengan nama file **latih1.php**, kemudian jalankan pada browser!

Sesi II

Konfigurasi PHP

Durasi

1 jam

Jenis

Teori dan Praktik

Media

PHP 5, XAMPP, Web Browser.

Tujuan

Setelah membaca sesi ini, siswa diharapkan:

- ❖ Mampu melakukan instalasi PHP 5 pada sistem operasi yang mendukung.
- ❖ Mampu merancang perangkat keras yang dibutuhkan oleh PHP 5.

Pada sesi pertama telah diberikan pengantar PHP dan berbagai macam fasilitas, kemampuan, keunggulan, dan lain sebagainya dari bahasa PHP. Sesi ini akan membahas kebutuhan software dan hardware untuk menginstal PHP 5 pada sistem operasi.

2.1 KEBUTUHAN PERANGKAT KERAS

PHP dapat berjalan pada berbagai sistem operasi berbasis Microsoft Windows 95/98/ME/NT/2000/XP/VISTA, dan sistem operasi lain, seperti: UNIX, LINUX, Solaris, dan Open Suse. Kebutuhan perangkat keras juga menyesuaikan kebutuhan sistem operasi yang digunakan, karena PHP memiliki ukuran file kecil dan bersifat *multi-platform*.

Varian sistem operasi tersebut dapat digunakan jika dalam fungsinya secara *stand alone*. Dengan demikian, jika dihendaki pada akhirnya akan dilakukan *publishing* atau *upload* terhadap web yang telah dibangun dengan menggunakan PHP, maka sistem operasi yang dibutuhkan harus mendukung web server. Sebagai contoh, untuk sistem operasi berbasis Windows dapat menggunakan Windows NT/2000 atau di atasnya:

Spesifikasi atau kebutuhan hardware yang dibutuhkan untuk instalasi dengan Windows NT adalah:

- Processor 486/33 MHz atau yang lebih tinggi.
- Memori atau RAM minimal 16 MB.
- Ruang kosong hardisk minimal sekitar 125 MB.

Informasi selengkapnya mengenai konfigurasi hardware yang dibutuhkan untuk instalasi pada Windows NT dapat dilihat di alamat: <http://www.microsoft.com/ntserver/>.

Konfigurasi hardware untuk instalasi pada Windows 2000 adalah:

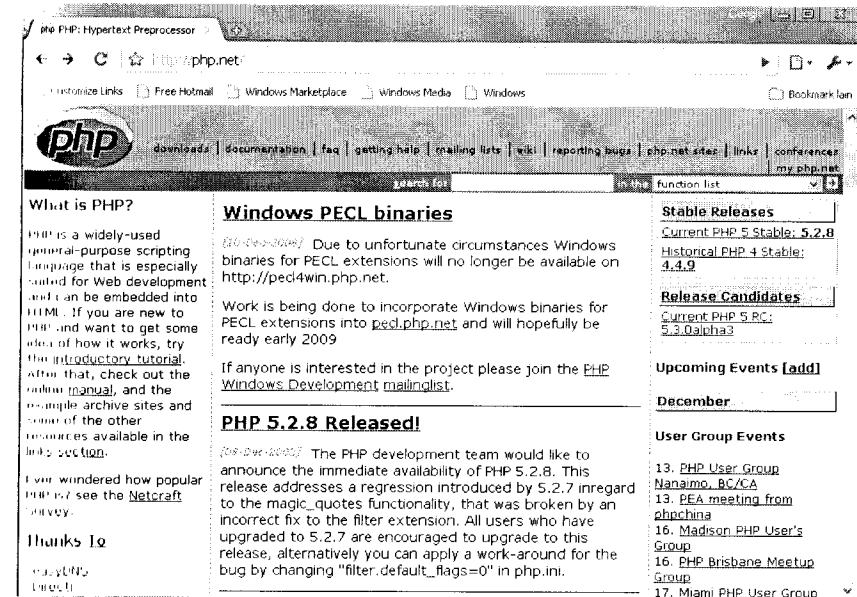
- Processor minimum Pentium 133 MHz atau yang lebih tinggi.
- RAM minimal 64 MB.
- Ruang kosong hardisk 650 MB, dengan kapasitas harddisk minimal 2 GB.

2.2 KEBUTUHAN PERANGKAT LUNAK

Selain perangkat keras, kebutuhan perangkat lunak yang tergantung pada sistem operasi yang digunakan juga merupakan syarat utama yang tidak kalah pentingnya. Perangkat lunak yang dibutuhkan pada peng-

gunaan sistem operasi Windows ME atau Windows 9x adalah Personal Web Server (dapat dilihat pada paket CD sistem operasi). Jika tidak ada, Anda dapat melihatnya pada paket CD Windows 98 Second Edition Full Release. Untuk Windows XP dan Windows Vista, PHP dapat dijalankan dengan melakukan setting IIS terlebih dahulu.

Pada Windows NT dan Windows 2000 dapat diinstal *service pack* yang berisi Internet Information Services (IIS). Selain software tersebut, Anda juga membutuhkan *engine* yang dapat di-download pada alamat situs: <http://www.php.net/>, seperti yang terlihat pada Gambar 2.1.



Gambar 2.1 Situs php.net

jika sistem operasi yang digunakan memiliki platform UNIX, maka Anda perlu menambahkan modul Apache yang dapat diambil dari <http://www.apache.org>, mengingat UNIX belum memiliki *engine* untuk menjalankan PHP.



Gambar 2.2 Apache Server

Setelah melakukan download file php dari situsnya, ikuti beberapa langkah instalasi PHP 5 dari *source code* sebagai berikut:

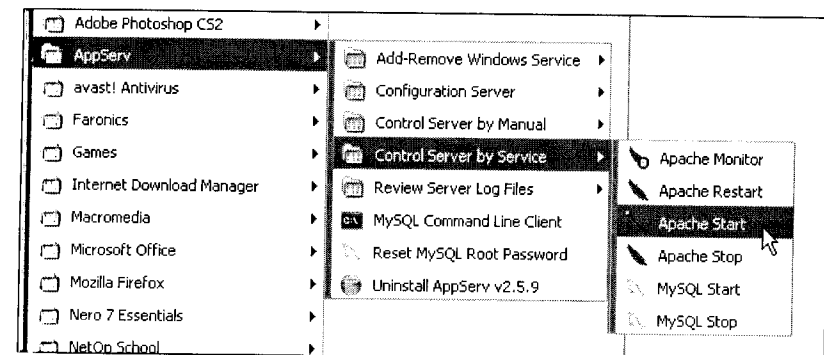
- **Instalasi Apache**
 1. Download dari <http://httpd.apache.org>, misalnya: apache 1.3.37.
 2. Uncompress di sembarang direktori `bash% tar zxvf apache_1.3.37.tar.gz`.
 3. Konfigurasi dan kompilasi `bash% cd apache_1.3.37`
`bash% ./configure --prefix=/di r/ in stal as i/ ap ach e/ and a enableshared=max`
`bash% make`.
 4. Instal: `bash% make install`.
 5. Instalasi Apache selesai.
- **Instalasi module PHP dan PHP CLI:**
 1. Download dari situs: www.php.net.
 2. Uncompress pada sembarang direktori `bash% tar zxvf php 5.2.3.tar.gz`.
 3. Konfigurasi dan kompilasi.

Untuk membuat suatu situs yang bersifat dinamis dengan dukungan database, Anda membutuhkan software dan tool sebagai berikut:

- Sebuah aplikasi web server (Apache, Abyss, dan IIS).
- PHP.
- MySQL.
- Sebuah web browser (Microsoft Internet Explorer, Mozilla Firefox, Safari, dan lain sebagainya).
- Sebuah teks editor untuk PHP yang mendukung aplikasi WYSIWYG seperti produk dari Adobe Dreamweaver beserta IDE (Integrated Development Environment).

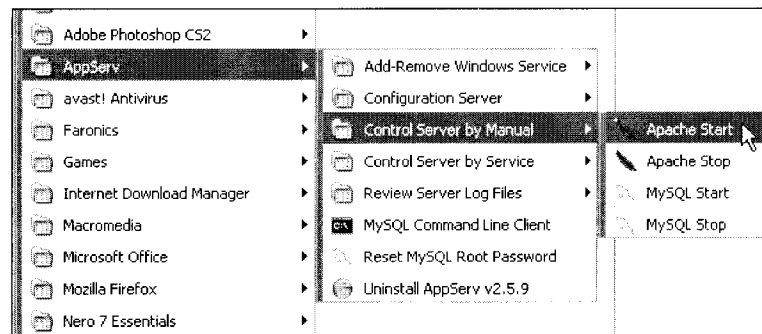
2.3 MENGAKTIFKAN PHP 5

Setelah proses instalasi selesai, lanjutkan dengan menjalankan PHP. PHP Iriad pada Start Menu tidak beralamatkan AppServ, melainkan pada PHP Iriad karena tidak membutuhkan dukungan server dari AppServ. PHP Iriad sudah menjadi satu paket termasuk Database MySQL.



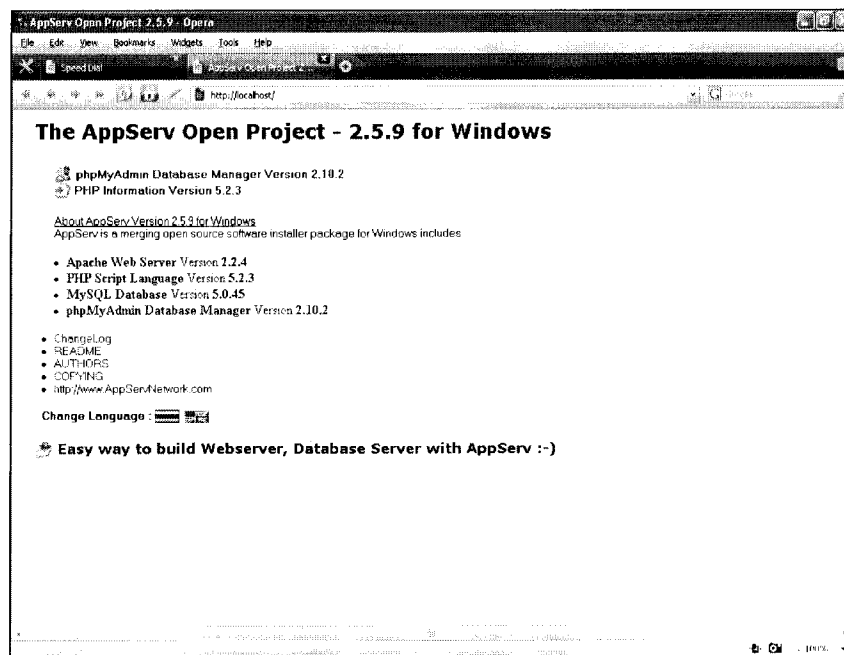
Gambar 2.3 Menjalankan Apache Server dengan Service Server

Untuk menjalankan PHP 5, klik menu **Start > All Programs > AppServ > Control Server by Service > Apache Start** hingga muncul ikon Apache pada system tray yang menandakan bahwa PHP telah diaktifkan.



Gambar 2.4 Menjalankan Apache Server dengan Manual Server

Setelah Control Server by Service diaktifkan, jalankan mode Control Server by Manual dan pilih Apache Start. Setelah selesai, gunakan web browser untuk membuka tampilan utama AppServ dengan mengetikkan alamat pada address bar "http://localhost/".



Gambar 2.5 Apache Server dijalankan pada Internet Explorer



Pada saat menjalankan localhost menggunakan sistem operasi Windows (XP, Vista, dan lainnya), maka akan muncul halaman bawaan Windows. Nonaktifkan Internet Information Services, kemudian gunakan port selain 80 untuk AppServs karena port tersebut sudah digunakan oleh Windows.

PHP Version 5.2.3

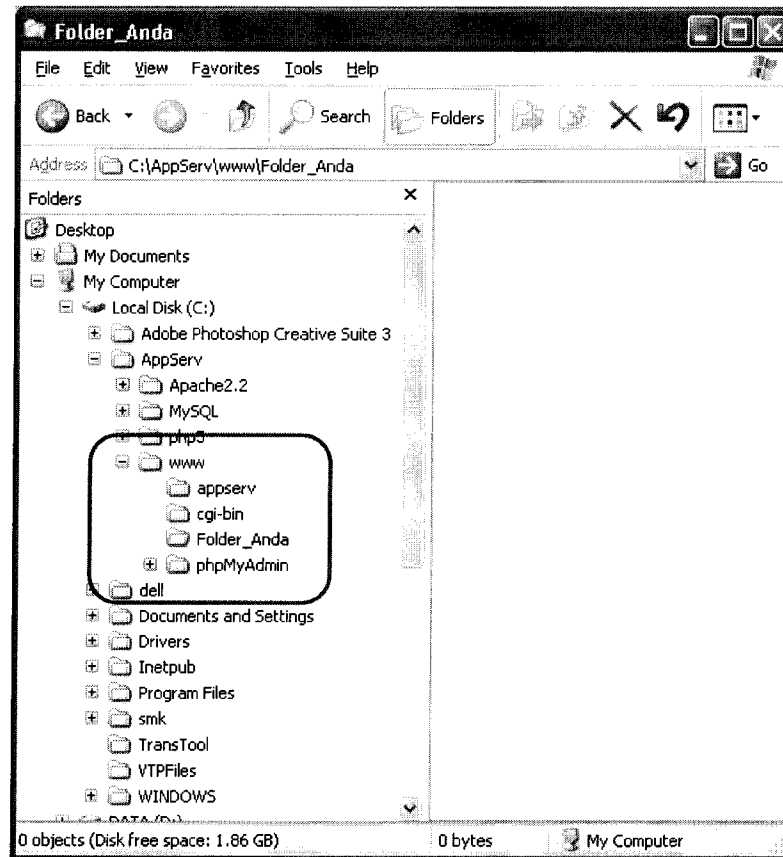
System	Windows NT TEACHER 5.1 build 2600
Build Date	May 31 2007 09:36:39
Configure Command	cscript /nologo configure.js "--enable-snapshot-build" "--with-gd=shared"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\WINDOWS
Loaded Configuration File	C:\WINDOWS\php.ini
PHP API	20041225
PHP Extension	20060613
Zend Extension	220060519
Debug Build	no
Thread Safety	enabled
Zend Memory Manager	enabled
IPv6 Support	enabled
Registered PHP Streams	php, file, data, http, ftp, compress.zlib
Registered Stream Socket Transports	tcp, udp
Registered Stream Filters	convert.iconv.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, zlib.*

This program makes use of the Zend Scripting Language Engine:
 Zend Engine v2.2.0, Copyright (c) 1998-2007 Zend Technologies

Powered By

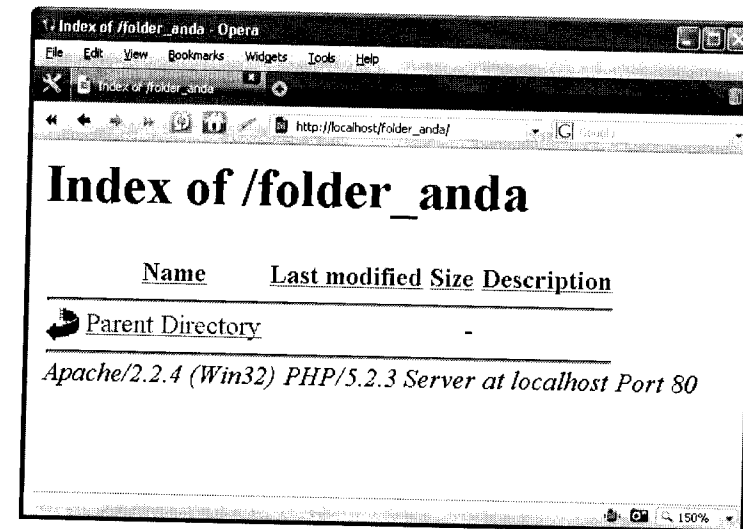
Gambar 2.6 PHP Version 5.2.3 Information

Pada halaman PHP Information, Anda dapat melihat properties PHP yang digunakan. Informasi yang ditampilkan meliputi: kapan PHP versi tersebut dirilis hingga fitur-fitur penunjang yang terdapat pada PHP versi tersebut.



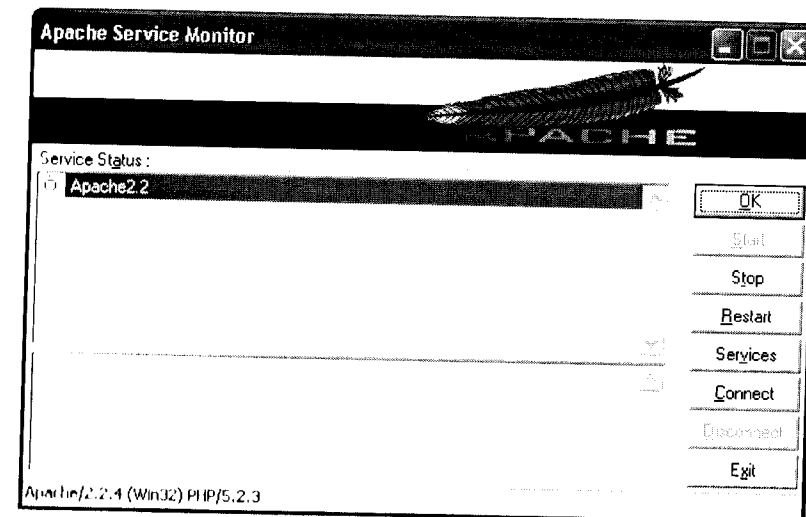
Gambar 2.7 Folder Penyimpanan Dokumen PHP pada Windows Explore

Gambar 2.7 menunjukkan folder penyimpanan dokumen PHP. Dengan demikian, jika Anda menyimpan dokumen PHP - selain pada folder tersebut - maka hasil eksekusi dari skrip yang dibuat tidak akan muncul pada browser.



Gambar 2.8 Tampilan Dokumen pada Windows Explorer melalui Internet Explorer

Setelah menyimpan dokumen PHP di dalam direktori yang telah ditentukan, Anda dapat membuka dokumen tersebut dalam browser. Pada address bar ketikkan alamat berikut: "http://localhost/ Folder_Anda". Halaman tersebut akan menampilkan isi direktori file PHP Anda.



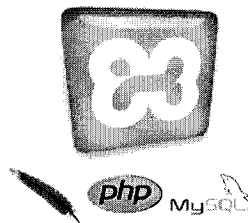
Gambar 2.9 Apache Service Monitor pada System Tray

Apache Service Monitor digunakan untuk mengetahui status Apache, apakah sudah aktif. Jika belum, Anda dapat mengaktifkannya melalui menu **Start** pada Windows.

2.4 XAMPP

Bagi sebagian programmer berpengalaman, menginstal web server Apache tidak mudah dan menyulitkan saat user ingin menambah MySQL, PHP, dan Perl. Namun demikian, XAMPP dapat menjawab dan mengatasi semua permasalahan tersebut. Dilengkapi dengan Control Panel berbasis GUI, PHPMyAdmin, dan *add-ons* yang mendukung, XAMPP bisa dijadikan sebagai web server dan database server serta pendukung PHP. Untuk mendownload XAMPP tidaklah sulit, Anda bisa mendapatkannya secara gratis di www.apachefriends.org/en/xampp.html.

XAMPP adalah salah satu paket instalasi Apache, PHP, dan MySQL secara instan yang dapat digunakan untuk membantu proses instalasi ketiga produk tersebut sama seperti PHPTriad. Selain paket instalasi instan, XAMPP juga memberikan fasilitas pilihan penggunaan PHP 4 atau PHP 5. Untuk melakukan migrasi ke versi lebih tinggi juga sangat mudah dilakukan dengan bantuan PHP-Switch yang telah disertakan oleh XAMPP. Sama halnya dengan PHP, XAMPP bersifat *free* atau gratis untuk digunakan.



Gambar 2.10 XAMPP

Selain sebagai PHP Server, XAMPP juga memiliki *mod_perl* yang berfungsi sebagai ASP Server, sehingga Anda bisa menggunakan XAMPP dengan bahasa pemrograman ASP.

XAMPP adalah suatu langkah mudah untuk menginstal distribusi Apache yang memiliki fitur program MySQL, PHP, dan Perl. XAMPP juga sangat mudah diinstal dan digunakan. Anda cukup mendownload, mengekstrak, dan mulai menggunakannya. XAMPP berdasar pada lisensi GNU General

Public License, sehingga instalasi XAMPP tidak memerlukan perubahan pada register Windows dan tidak perlu mengedit file konfigurasinya. Terdapat program *default* XAMPP yang ditulis dalam bahasa PHP menggunakan MySQL, dan aplikasi buku tamu yang ditulis dalam bahasa Perl, serta beberapa *utility* demo program lainnya.

Dalam paket XAMPP, Anda akan memperoleh beberapa fitur sebagai berikut:

- Apache
- Cgi-Bin
- PHP
- MySQL
- FTP
- Mercury Mail (SMTP)
- PHP MyAdmin
- Perl
- Webalizer
- dll

Pada XAMPP terdapat 4 macam distribusi, yakni:

- **XAMPP pada Windows.** Untuk sistem operasi Windows, seperti untuk Windows 98, NT, 2000, 2003, XP, and Vista. XAMPP versi Windows ini memiliki fitur Apache, MySQL, PHP + PEAR, Perl, *mod_php*, *mod_perl*, *mod_ssl*, OpenSSL, *phpMyAdmin*, Webalizer, Mercury Mail Transport System for Win32 and NetWare Systems v3.32, Ming, JpGraph, FileZilla FTP Server, *mcrypt*, *eAccelerator*, *SQLite*, dan *WEB-DAV + mod_auth_mysql*.
- **XAMPP pada Solaris.** Untuk platform Solaris (baru dikembangkan dan diuji pada sistem operasi Solaris 8 dan Solaris 9) yang meliputi: Apache, MySQL, PHP & PEAR, Perl, ProFTPD, *phpMyAdmin*, OpenSSL, *Freetype2*, *libjpeg*, *libpng*, *zlib*, *expat*, Ming, Webalizer, dan *pdf class*.
- **XAMPP pada Mac OS X.** Untuk platform Mac OS X, meliputi: Apache, MySQL, PHP & PEAR, *SQLite*, Perl, ProFTPD, *phpMyAdmin*, OpenSSL, GD, *Freetype2*, *libjpeg*, *libpng*, *zlib*, Ming, Webalizer, *mod_perl*, *eAccelerator*, *phpSQLiteAdmin*. Untuk distribusi XAMPP Mac OS X ini barulah sampai ke tahap pengembangan, belum sampai ke final.

- **XAMPP pada Linux.** Untuk sistem operasi Linux (telah diuji untuk sistem operasi SuSE, RedHat, Mandrake dan Debian), meliputi: Apache, MySQL, PHP & PEAR, Perl, ProFTPD, phpMyAdmin, OpenSSL, GD, Freetype2, libjpeg, libpng, gdbm, zlib, expat, Sablotron, libxml, Ming, Webalizer, pdf class, ncurses, mod_perl, FreeTDS, gettext, mcrypt, mhash, eAccelerator, SQLite, dan IMAP C-Client.

2.5 RANGKUMAN

Karena bersifat *freeware*, banyak programmer yang bermigrasi ke PHP dari bahasa pemrograman berbayar. Logika pemrograman juga lebih mudah karena hampir sama dengan bahasa pemrograman pada umumnya. Selain itu, sifat PHP yang *multi-platform* menjadikan PHP populer di dunia *server side*.

2.6 KUIS

1. Sebutkan kebutuhan perangkat keras untuk PHP 5!
2. Sebutkan kebutuhan perangkat lunak untuk PHP 5!
3. Sebutkan spesifikasi minimum untuk instalasi PHP pada Windows XP!
4. Adakah perbedaan konfigurasi software untuk PHP 5 pada platform Windows NT dan 9x? Jelaskan!
5. Di mana Anda harus menyimpan dokumen PHP?

2.7 LATIHAN

Carilah software PHP, mulai dari versi yang pertama sampai versi PHP 5, kemudian perhatikan ukuran file aplikasi tersebut. Instal aplikasi tersebut secara berkala, kemudian sebutkan wujud perkembangan PHP itu sendiri. Menurut Anda, dengan ukuran file yang berbeda terdapat fasilitas-fasilitas yang ditambahkan atau bahkan mungkin dikurangi? Jelaskan!

Sesi III

Elemen dalam PHP

Durasi	2 jam
Jenis	Teori dan Praktik
Media	PHP 5, Web Browser.
Tujuan	Setelah membaca sesi ini, siswa diharapkan dapat: ❖ Memahami dasar pemrograman PHP. ❖ Mengerti berbagai elemen dalam PHP. ❖ Mengerti tipe data dan array dalam PHP. ❖ Menerapkan variabel pada PHP ke dalam server.

Script pemrograman PHP dapat ditanamkan dalam sintaks HTML. Kode PHP diawali dengan sintaks "<?" dan diakhiri dengan sintaks ">". File tersebut diberi akan berekstensi **.php**. Pada beberapa server yang mendukung, blok scripting PHP dapat diawali dengan <? dan diakhiri dengan >. Agar lebih kompatibel, sebaiknya menggunakan bentuk yang standar (<?php >). Saat file tersebut diakses, server akan mendeteksi jika file tersebut mengandung sintaks PHP. Server kemudian menerjemahkan kode tersebut dan menghasilkan output berupa sintaks HTML, yang kemudian akan dikirim ke browser user yang mengakses web dari web server. Sebagaimana bahasa pemrograman umumnya, PHP memiliki sintaks program, jenis variabel, definisi variabel, operator, struktur kontrol, dan fungsi.

3.1 DASAR PEMROGRAMAN PHP

Skrip PHP disimpan sebagai *plain text* dalam format ASCII, yang memungkinkan skrip PHP dapat ditulis di hampir semua editor text, seperti: notepad, wordpad, dan lain-lain. Skrip PHP adalah kode yang disertakan di sebuah halaman HTML dan kode tersebut dijalankan oleh server yang akan dieksekusi sebelum dikirim ke browser.

Contoh file PHP (contoh.php):

```
<html>
<?
Print ("PHP Terpasang dalam HTML");
?>
</html>
```

Pada file **.html**, HTTP server hanya melewatkan *content* dari file menuju ke browser. Server tidak mencoba untuk mengerti atau memroses file karena itu adalah tugas sebuah browser untuk melakukan kompilasi hasil program.

File berekstensi **.php** akan ditangani secara berbeda -file yang memiliki kode PHP akan diperiksa. Web server akan mulai bekerja apabila berada di luar lingkungan kode HTML. Oleh karena itu, server akan melewati semua *content* yang berisi kode HTML, CSS, JavaScript, simple text di browser tanpa diinterpretasikan di server.

Penulisan skrip PHP memiliki aturan sebagai berikut:

- Diawali <?Php dan diakhiri ?> atau diawali <? dan diakhiri ?>
- <script language="php"> dan </script>
- <% dan %>
- Skrip PHP bersifat *case-sensitive*, sehingga penulisan huruf besar dan huruf kecil pada skrip PHP harus diperhatikan. Untuk nama identifier yang dibuat oleh programmer (berupa variabel, konstanta, fungsi, dll), sedangkan yang *noncase-sensitive* untuk *built-in* dari PHP, seperti:

\$joko ≠ \$Joko ≠ \$JOKO

\$hitungPanjang() ≠ \$hitungPanjang()

while ≠ WHILE

echo ≠ ECHO

Penulisan kode-kode PHP dapat dilakukan dengan banyak cara, seperti:

<?Php

```
echo("Selamat Belajar PHP Bersama Andi dan Wahana");
```

>?

Atau, dapat juga ditulis:

<?

```
echo("Selamat Belajar PHP Bersama Andi dan Wahana");
```

>?

Bagaimana membedakan skrip PHP dengan HTML atau skrip jenis lainnya (JavaScript dan VBScript)?

Suatu skrip akan dikenali sebagai skrip PHP bila diapit oleh tanda:

```
<?php ....?>
```

```
<?. ....?>
```

```
<script language="PHP">...</script>
```

Sekarang, buatlah skrip PHP dengan tanda apit yang lain, seperti skrip:

```
<script language="PHP">
```

```
echo ("Mudahnya Belajar PHP " ");
```

```
</script>
```

Atau:

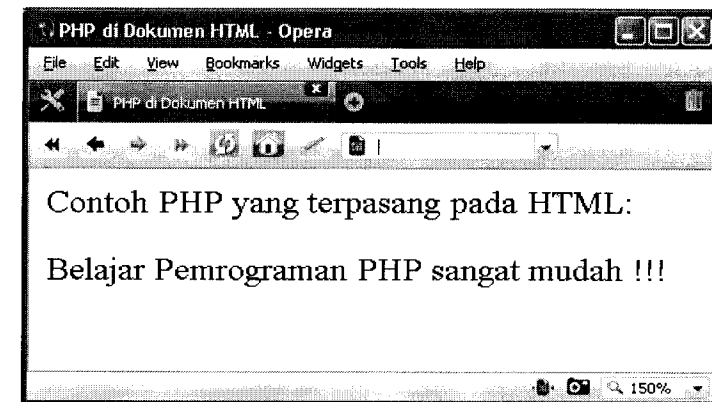
```
<?
echo ("Mudahnya Belajar PHP!! ");
?>
```

Perintah-perintah PHP dapat disisipkan di bagian manapun dalam dokumen HTML; yang perlu dilakukan hanyalah menyisipkannya di skrip HTML dan menyimpannya sebagai ***.php**.

Cobalah buat skrip berikut:

```
<HTML>
<HEAD>
<TITLE>Menyisipkan PHP di Dokumen HTML</TITLE>
</HEAD>
<BODY>
Contoh PHP yang terpasang pada HTML: <BR>
<P>
<?php
echo("Belajar Pemrograman PHP sangat mudah !!!");
?>
</BODY>
</HTML>
```

Jika dijalankan pada web browser, maka akan muncul tampilan seperti pada **Gambar 3.1**.



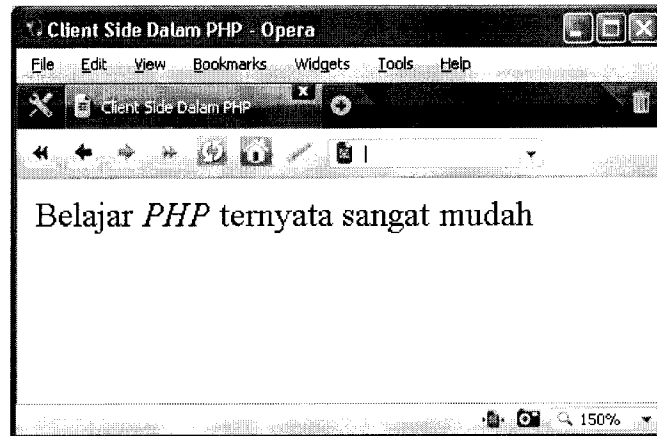
Gambar 3.1 Skrip PHP yang Terpasang dengan HTML

3.1.1 PHP DAN CLIENT SIDE CODE

Skrip PHP dapat dikembangkan dalam dokumen bersifat dinamis berbasis Client Side, seperti contoh program berikut:

```
<html>
<head>
<title>Client Side Dalam PHP</title>
</head>
<body>
<?php
echo ("Belajar <EM>PHP</EM> ternyata sangat mudah");
?>
</body>
</html>
```

Saat dijalankan pada browser, tulisan PHP akan dicetak miring.



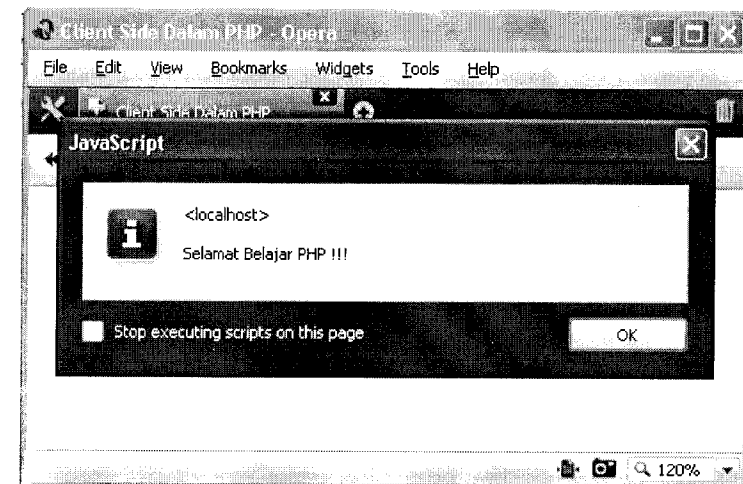
Gambar 3.2 Skrip PHP Berbasis Client Side

Penggunaan tag `<EM>` **Emphasis** diletakkan di sekitar teks "PHP". Webserver dan PHP akan membaca bahwa kata tersebut hanya merupakan bagian dari teks, yang kemudian dikirimkan ke browser.

Skrip PHP juga dapat disisipkan dengan skrip JavaScript, seperti yang terlihat pada contoh berikut:

```
<html>
<head>
<title>Client Side Dalam PHP</title>
</head>
<body>
<?php
echo ("<script language='javascript'>alert ('Selamat Belajar PHP !!!');
</script>");
?>
</body>
</html>
```

Pada saat dijalankan pada browser, maka akan muncul tampilan pesan peringatan, seperti yang ditunjukkan pada **Gambar 3.3**.



Gambar 3.3 Penyisipan Javascript dalam Sintaks PHP

Berdasarkan contoh pada **Gambar 3.3**, pada saat webserver mengeksekusi statement `echo`, maka akan mengirimkan pesan ("`<script language='javascript'>alert ('Selamat Belajar PHP !!!'); </script>`" ke dalam browser. Skrip tersebut akan diterjemahkan dalam kode JavaScript. Browser akan membaca dan menampilkan pesan peringatan (Box Alert).

3.2 STATEMENT, EXPRESSION, DAN COMMENT

Suatu instruksi hanya dapat dipisahkan oleh tanda *semicolon* (;). Dengan demikian, maka setiap instruksi yang panjang, seperti mendefinisikan/mencetak suatu teks, dapat dilakukan pada beberapa baris.

3.2.1 STATEMENT

Statement merupakan satuan perintah dalam PHP, yang wajib diakhiri dengan tanda *semicolon*/titik-koma (;).

Contoh statement:

```
echo("hello user ...!");
```

Contoh lainnya:

```
echo("4 + 5 = " . 4+5);
```

3.2.2 EXPRESSION

Expression adalah satu bagian kecil kode yang akan dihitung hasilnya oleh php.

Contoh Expression:

6 + 8

Penggunaan Expression:

```
echo("6 + 8 = " . 6+8);
```

3.2.3 COMMENT

Comment adalah bagian kode yang tidak dieksekusi/dijalankan, yang dibuat untuk memperjelas atau memberi keterangan pada kode program.

Dalam PHP, komentar program bisa menggunakan:

- Diawali `/*` dan diakhiri `*/`
- Diawali `//`
- `#`

Contoh penggunaan:

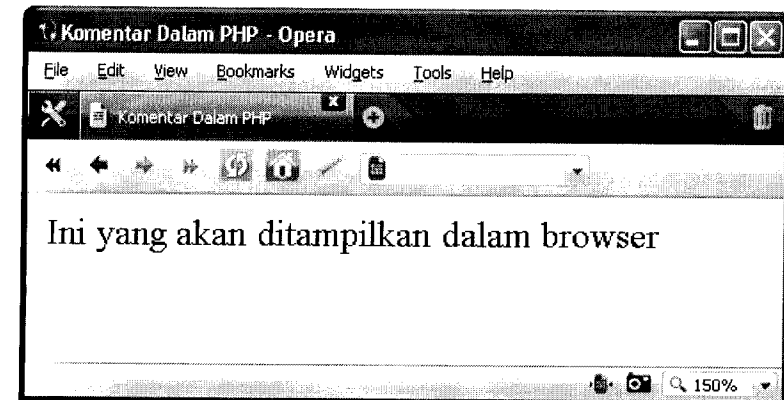
```
echo("9 + 6 = " . 9+6); // menampilkan hasil 9 + 6
```

Comment dengan banyak baris dibuat dengan pasangan `/*` dan `*/`. Semua tulisan yang dibuat di antara tanda tersebut tidak akan dieksekusi oleh interpreter. Contoh penggunaan komentar:

```
<html>
<head>
<title>Komentar Dalam PHP</title>
</head>
<body>
<?

```

```
/* Komentar ini tidak akan muncul dalam browser */
// Untuk yang ini komentar satu baris
# Yang ini juga komentar
echo "Ini yang akan ditampilkan dalam browser";
?>
</body>
</html>
```



Gambar 3.4 Contoh Penggunaan Komentar dalam PHP

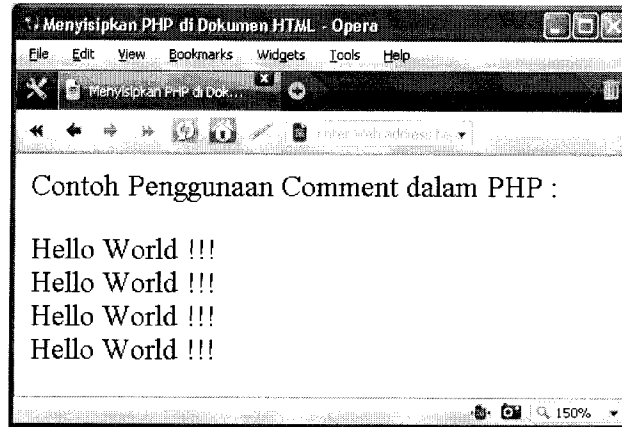
Contoh program:

- `HTML>`
- `HEAD>`
- `TITLE>Menyisipkan PHP di Dokumen HTML</TITLE>`
- `/HEAD>`
- `BODY>`

Contoh Penggunaan Comment dalam PHP : `<BR>`

```
• ?php
print "Hello World !!!<br>"; //Ini Komentar
echo "Hello World !!!<br>"; #Ini juga Komentar
echo ("Hello World !!!<br>"); /* Ini juga Komentar */
```

```
printf("Hello World !!!<br>");
?>
</BODY>
</HTML>
```



Gambar 3.5 Contoh Penggunaan Komentar dalam PHP

3.3 TIPE DATA

Seperti halnya bahasa pemrograman pada umumnya, PHP juga mengenal beberapa tipe data. Tipe data tersebut merupakan sebuah variabel yang dapat ditentukan secara otomatis oleh PHP, bergantung pada operasi yang sedang dilakukan menggunakan variabel tersebut. Ada tiga tipe data dasar pada PHP, yaitu: integer, real (floating point), dan string. Selain itu, ada juga tipe data -yang tidak termasuk dalam tipe data dasar-, yaitu Array Object, Boolean, serta Type Jugling. Pada umumnya, PHP memiliki delapan tipe data, antara lain:

- Integer
- Floating Point
- String
- Array
- Objects
- Resource
- NULL
- Boolean

3.3.1 TIPE DATA PHP

Seperti sudah disebutkan sebelumnya, PHP memiliki berbagai macam tipe data yang memiliki nilai pada variabel yang menyertai tipe data tersebut.

Integer

Tipe data integer yang meliputi semua bilangan bulat, besar range data integer pada PHP sama dengan range tipe data long pada bahasa C, yaitu sekitar -2.147.483.648 sampai 2.147.483.647 pada *platform* 32bit, dan menggunakan 4 bytes dari memori.

PHP akan mengonversi data integer menjadi *floating point* secara otomatis jika berada di luar range tersebut. Integer dapat dinyatakan dengan bentuk octal, desimal, dan heksadesimal.

Contoh:

```
$biaya = 500;
$a = 1234; (bilangan desimal)
$a = -1234; (bilangan negatif)
$a = 0123; (bilangan oktal)
$a = 0x1F; (bilangan heksadesimal)
```

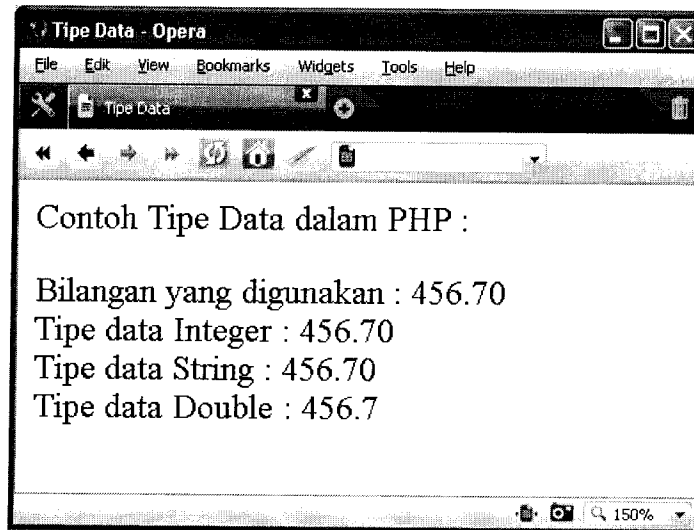
Contoh program:

```
<HTML>
<HEAD>
<TITLE>Tipe Data</TITLE>
</HEAD>
<BODY>
Contoh Tipe Data dalam PHP : <BR>
<P>
<?php
$Angka="456.70";
print("Bilangan yang digunakan : $Angka<BR> \n");
```

```
// Penggunaan tipe data integer
print("Tipe data Integer : $Angka<BR> \n");

// Penggunaan tipe data string
settype($Angka, "String");
print("Tipe data String : $Angka<BR> \n");

//Penggunaan tipe data double
settype($Angka, "Double");
print("Tipe data Double : $Angka<BR> \n");
?>
</BODY>
</HTML>
```



Gambar 3.6 Tipe Data

Floating Point (Bilangan Riil)

Bilangan **Floating Point** (floats, double, dan real) merupakan representasi dari bilangan desimal atau pecahan. Range tipe data *floating point* sama dengan range tipe data double pada bahasa C, yaitu: 1.7E308 sampai 1.7E308.

Floating point atau bilangan riil dapat dinyatakan sebagai berikut:

\$a = 1.234;

\$a = 1.2e3;

\$a = 7E-10;

Boolean

Tipe data Boolean merupakan tipe data yang hanya memiliki nilai TRUE dan FALSE. Nilai Boolean biasanya digunakan pada struktur kontrol program, seperti *if* atau *if else*. PHP hanya mengenal dua *keyword*, yaitu: *true* dan *false*. *True* merepresentasikan nilai integer '1' dan *False* merepresentasikan string kosong. *Keyword true* dan *false* tidak *case-sensitive*.

Ada beberapa kondisi dalam nilai Boolean harus digunakan, yaitu:

- Untuk integer dan floating point, nilai Boolean adalah false jika nilainya 0, dan lainnya bernilai true.
- Untuk variabel yang belum dapat dibuat atau didefinisikan nilainya adalah false. Untuk tipe data object, bernilai false jika tidak ada variabel atau fungsi/method yang terdefinisi di dalamnya; jika sebaliknya, maka bernilai true.
- Untuk string, nilai boolean-nya false jika string kosong (""), sisanya bernilai true.
- Pada tipe data Array, akan bernilai false jika elemennya kosong dan bernilai true jika sebaliknya.

String

String atau teks adalah deretan karakter. Dalam PHP, karakter mempunyai ukuran memori 1 byte dan mempunyai 256 karakter yang berbeda. String dapat dibuat dengan menuliskan teks di antara tanda kutip tunggal (') atau tanda kutip ganda (").

\$a = "Learn PHP";

```
$a = 'Belajar PHP';
$nama_depan='Joko Riyanto';
$hari='12 Oktober 2008';
```

Penulisan string dapat dilakukan dengan menggunakan *echo()* atau *print()*. Untuk mencetak nilai string pada browser, Anda dapat menuliskan skrip sebagai berikut:

```
echo "Selamat Datang, $nama_depan";
```

Jika Anda memberikan nilai suatu variabel (*jenis\$buku*), kemudian nilai baru akan ditulis melebihi jumlah dari nilai sebelumnya. Seperti contoh:

```
$buku = 'Komputer';
$buku = 'Desain Grafis';
/* $buku sekarang mempunyai nilai dari
'Desain Grafis'. */
```

PHP tidak memiliki batas seberapa besar nilai string yang dibuat. Namun secara teori, Anda dapat memberikan batasan sesuai dengan kapasitas pada server.

Perbedaannya, jika menggunakan tanda petik tunggal, maka pada string itu tidak dapat dimasukkan variabel dan *escape sequence handling*. Sebuah literal string dapat dituliskan dalam tiga cara, yaitu:

- Single quoted
- Double quoted
- Herodoc syntax

Single Quoted

Ekspresi string yang paling mudah adalah dengan Single Quoted, yaitu dengan menggunakan tanda (*'*), sehingga saat dikehendaki mencetak tanda (*'*) harus menambahkan tanda *backslash* (**).

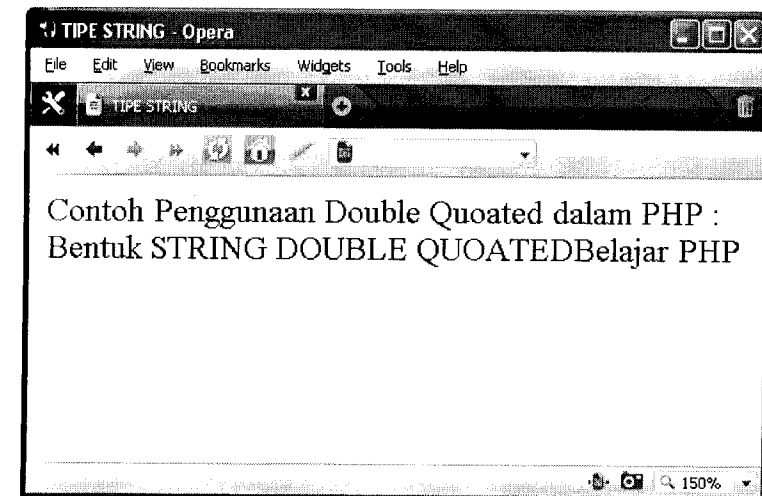
Contoh program:

```
<HTML>
<HEAD>
<TITLE>TIPE STRING</TITLE>
</HEAD>
```

```
<BODY>
```

Contoh Penggunaan Single Quoted dalam PHP : *
*

```
<?php
echo 'Bentuk STRING SINGLE QUOTED';
echo 'Belajar PHP';
?>
</BODY>
</HTML>
```



Gambar 3.7 Bentuk STRING SINGLE QUOTED

Double Quoted

Pada STRING yang menggunakan Double Quoted (*"*), PHP dapat mengeksekusi bahwa sintaks yang diapit dengan tanda (*"*) merupakan karakter-karakter khusus.

```
<HTML>
<HEAD>
<TITLE>TIPE STRING</TITLE>
</HEAD>
```

<BODY>

Contoh Penggunaan Double Quoted dalam PHP :

<?php

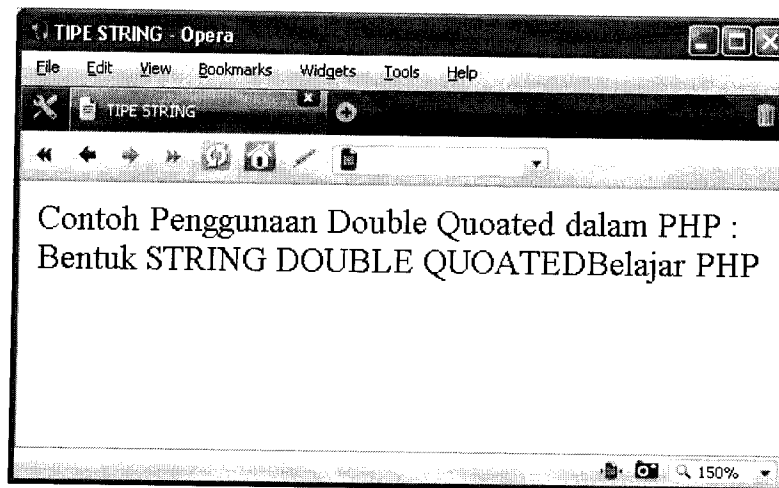
echo "Bentuk STRING DOUBLE QUOTED";

echo "Belajar PHP";

?>

</BODY>

</HTML>



Gambar 3.8 Bentuk STRING DOUBLED QUOTED

Pada dasarnya, browser yang dijalankan dengan STRING SINGLE QUOTED atau STRING DOUBLED QUOTED akan menghasilkan tampilan yang sama. Server akan membaca bahwa string tersebut merupakan penanda bahwa pada baris tersebut yang akan ditampilkan dalam browser.

Heredoc Syntax

Bentuk lain untuk mengekspresikan STRING adalah dengan Heredoc Syntax, yaitu dengan menggunakan tanda "<<<". Cara penulisan diawali dengan "<<<" kemudian diikuti dengan *identifier* dan diakhiri dengan *identifier* pembuka.

Contoh:

```
!Nama=<<<EOD
```

Contoh STRING

Menggunakan Heredoc Syntax

Pemrograman PHP

```
EOD;
```

Array

Salah satu tipe data yang terbilang unik adalah Array karena array dapat menjadi sebuah nilai array yang suatu waktu dapat dipanggil melalui variabelnya.

Konsep Array

Array merupakan sebuah data yang mengandung satu atau lebih data atau kumpulan data, dan dapat di-index berdasarkan *numeric* maupun *string* yang disimpan dalam satu variabel. Array atau jajaran merupakan variabel berindeks yang dapat dimanipulasi dengan mudah, seperti layaknya tabel dalam sebuah *spreadsheet*. Definisi nilai array dapat dilakukan dengan mencantumkan index dari array tersebut.

```
$a[0] = "abc";
```

```
$a[1] = "def";
```

Index array dapat juga berupa string (associative array), seperti:

```
$a['nama'] = "JOKO";
```

Array juga merupakan sekumpulan variabel yang berisi lebih dari 1 variabel. Sebagai contoh, Anda ingin memasukkan daftar nama teman. Daripada Anda menyatakan tiap nama teman dengan variabel **\$teman1**, **\$teman2**, **\$teman3**, ...dst, maka akan lebih baik dan efisien jika Anda menggunakan variabel **\$teman** yang berupa array.

Berbeda dengan variabel sebelumnya yang hanya menyimpan satu nilai, array dapat menampung lebih dari satu nilai. Sebagai contoh, Anda akan menyimpan data nama teman-teman Anda dalam satu variabel **\$nama**. Ada tiga nama teman yang akan dimasukkan ke dalam variabel tersebut. Gambaran umumnya, dapat definisikan sebagai berikut:

Joko	Riyanto	Ani
------	---------	-----

\$nama

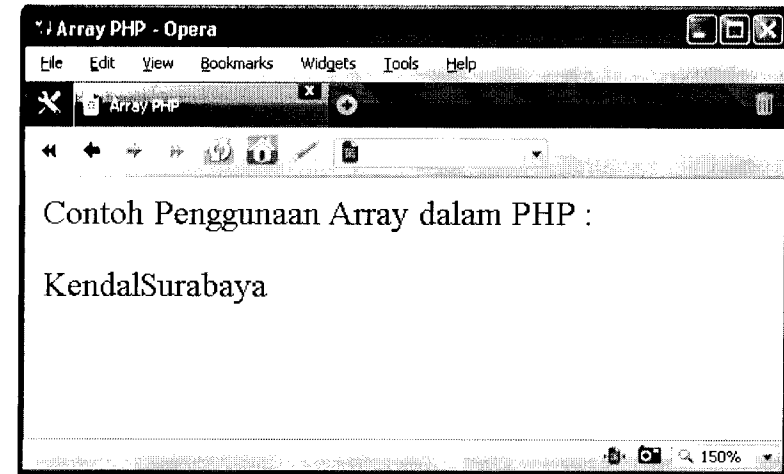
Tiap nilai dalam array (disebut elemen), diakses dengan menggunakan index. PHP mengenal dua macam index, yaitu: **index numerik** dan **index asosiatif**. Index numerik menggunakan angka untuk menandai tiap elemen, sedangkan index asosiatif memberi nama untuk masing-masing elemen.

Contoh program:

```
<HTML>
<HEAD>
<TITLE>Array PHP</TITLE>
</HEAD>
<BODY>
```

Contoh Penggunaan Array dalam PHP :


```
<P>
<?php
$arrKota = array("Semarang", "Kendal", "Jakarta", "Surabaya");
echo $arrKota[1]; //Kendal
echo $arrKota[3]; //Surabaya
?>
</P>
</BODY>
</HTML>
```



Gambar 3.9 Array dengan \$arrKota

Contoh skrip pada **Gambar 3.9** tersebut menggunakan Array untuk mencari nama-nama kota, dengan variabel **\$arrKota** sebagai penandanya, sebagai isi nilai dari array antara lain: array("Semarang", "Kendal", "Jakarta", "Surabaya"). Pada variabel array berjumlah 4, sehingga dapat diurutkan ke dalam kategori dengan menggunakan angka array("Semarang") = no 1, array("Kendal") = no 2, array("Jakarta") = no 3, dan array("Surabaya") = no 4. Untuk memanggil nilai suatu array tidak perlu menginputkan nama nilai tersebut cukup memasukkan angka.

Contoh program:

```
<HTML>
<HEAD>
<TITLE>Array PHP</TITLE>
</HEAD>
<BODY>
```

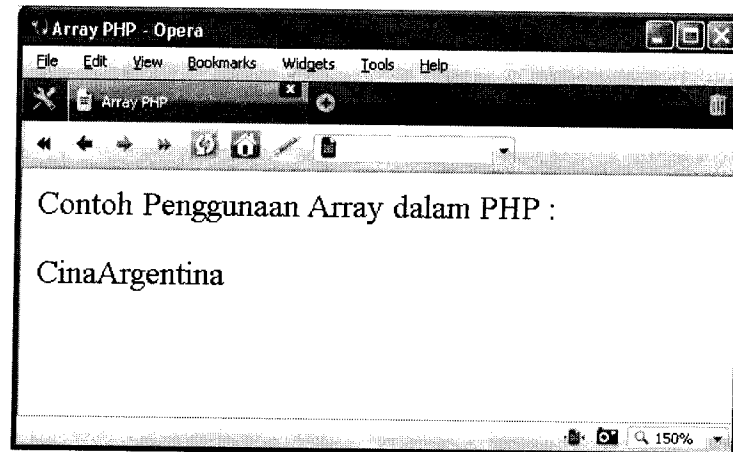
Contoh Penggunaan Array dalam PHP :


```
<P>
<?php
$arrNegara=array();
```

```

$arrNegara[] = "Indonesia";
$arrNegara[] = "Cina";
$arrNegara[] = "Argentina";
$arrNegara[] = "Filipina";
echo $arrNegara[1]; //Cina
echo $arrNegara[2]; //Argentina
?>
</P>
</BODY>
</HTML>

```



Gambar 3.10 Array dengan \$arrNegara

Berbeda dengan contoh skrip sebelumnya, Gambar 3.10 menunjukkan skrip dengan menggunakan variabel **\$arrNegara**, sedangkan untuk nilai dari yang dipanggil menggunakan variabel tersebut menggunakan penandaan:

```

$arrNegara[] = "Indonesia";
$arrNegara[] = "Cina";
$arrNegara[] = "Argentina";
$arrNegara[] = "Filipina";

```

Dalam pemanggilan nilai dari variabel **\$arrNegara** sama dengan yang terdapat pada contoh Gambar 3.10, yaitu dengan memasukkan angka. Angka tersebut akan memanggil beberapa variabel yang telah dideklarasikan sebelumnya.

Contoh program:

```

• HTML>
• HEAD>
• TITLE>Belajar PHP</TITLE>
</HEAD>
<BODY>

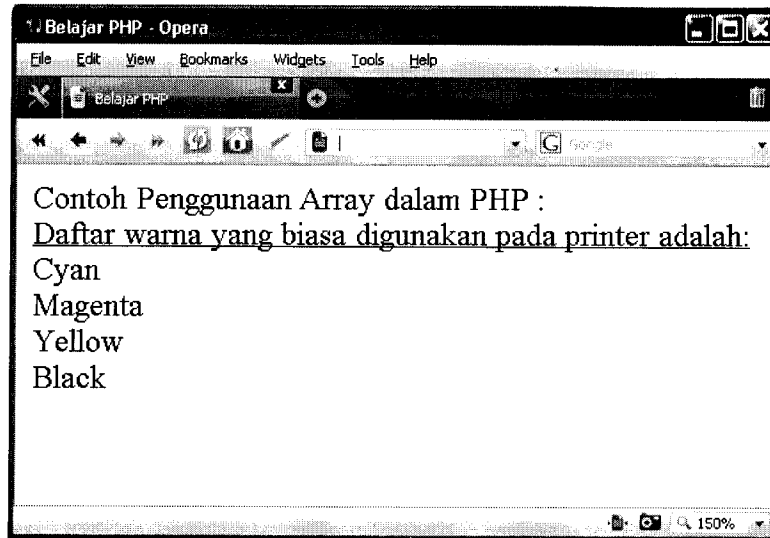
```

Contoh Penggunaan Array dalam PHP :


```

<?php
//Membuat Array
$warna=array("Cyan","Magenta","Yellow","Black");
echo("<U>Daftar warna yang biasa digunakan pada printer adalah:
</U><BR>");
//Mengakses Array
echo("$warna[0]<BR>");
echo("$warna[1]<BR>");
echo("$warna[2]<BR>");
echo("$warna[3]<BR>");
?>
</BODY>
</HTML>

```



Gambar 3.11 Array \$warna

Sintaks pada Gambar 3.11 menggunakan tipe STRING dengan cara pemanggilan nilai dalam variabel yang telah dideklarasikan terlebih dahulu, yaitu dengan memasukkan angka sesuai dengan urutan, seperti:

```
$warna=array("Cyan","Magenta","Yellow","Black");
```

Pada penggalan skrip di atas, variabel **\$warna** memiliki 4 nilai yang berisi komposisi warna. Sama halnya dengan:

```
$warna=array("0","1","2","3");
```

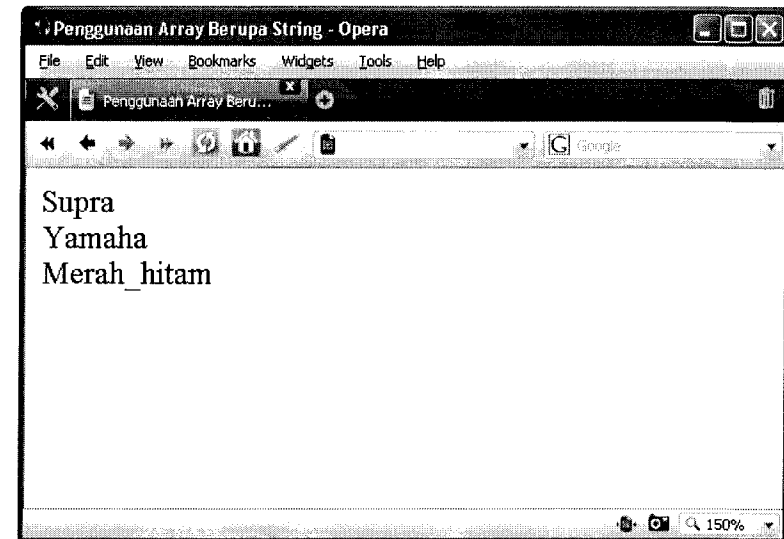
Jadi, untuk memanggil salah satu nilai (misalnya: Cyan) cukup masukkan angka "0", seperti contoh berikut: `echo("$warna[0]<BR>");`

Nilai dalam variabel array dimulai dari 0 desimal.

Contoh program:

```
<HTML>
<HEAD>
<TITLE>Penggunaan Array Berupa String</TITLE>
</HEAD>
<BODY>
```

```
· ?php
$motor["tipe"] = "Supra";
$motor["merk"] = "Yamaha";
$motor["warna"] = "Merah_hitam";
echo($motor["tipe"] . "<BR>\n");
echo($motor["merk"] . "<BR>\n");
echo($motor["warna"] . "<BR>\n");
?>
</BODY>
</HTML>
```



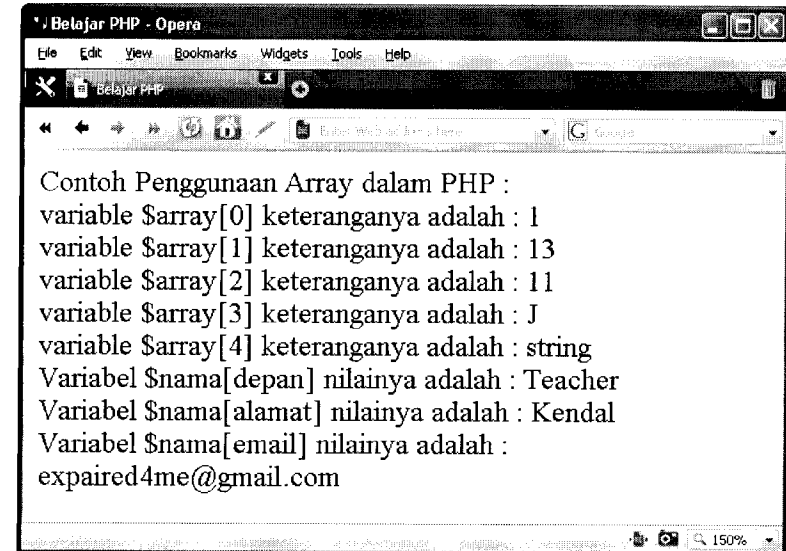
Gambar 3.12 Array \$armotor

Dalam hal ini, Array yang berupa STRING memanggil nilai pada variabel tidak dengan memasukkan angka, melainkan secara otomatis karena nilai yang dimasukkan tidak bersifat *optional*, sehingga apa yang menandai masing-masing nilai tersebut akan muncul menurut variabelnya.

Contoh program:

```
<HTML>
<HEAD>
<TITLE>Belajar PHP</TITLE>
</HEAD>
<BODY>
Contoh Penggunaan Array dalam PHP : <BR>
<?php
$array[]=1;
$array[]='13';
$array[]="11";
$array[]='J';
$array[]='string';
$nama[depan]="Teacher";
$nama[alamat]="Kendal";
$nama[email]="expired4me@gmail.com";
for ($a=0; $a<5; $a++)
{
echo ("variable \$array[$a] keteranganya adalah : $array[$a]<br>");
}
echo ("Variabel \$nama[depan] nilainya adalah : $nama[depan]<br>");
echo ("Variabel \$nama[alamat] nilainya adalah : $nama[alamat]<br>");
echo ("Variabel \$nama[email] nilainya adalah : $nama[email]<br>");
?>
</BODY>
</HTML>
```

Perhatikan skrip di atas, dengan memasukkan nilai, array tersebut dideklarasikan. Cara lain yang dapat dilakukan adalah dengan menggunakan fungsi *array()* atau *list()*. Dengan cara ini, angka index yang tidak dinyatakan akan diisi secara otomatis dengan angka index berikutnya, sesuai urutan pengisiannya. Dengan demikian, maka hasilnya tampak di tampilan browser pada **Gambar 3.13**.



Gambar 3.13 Array STRING dan Numeric

Pada contoh program di atas pemanggilan nilai dari suatu variabel *\$nama* dan *\$array* berupa numeric dan STRING. Untuk memasukkan kedua variabel ini ke dalam server, cukup memasukkan variabel yang telah dideklarasikan.

Contoh program:

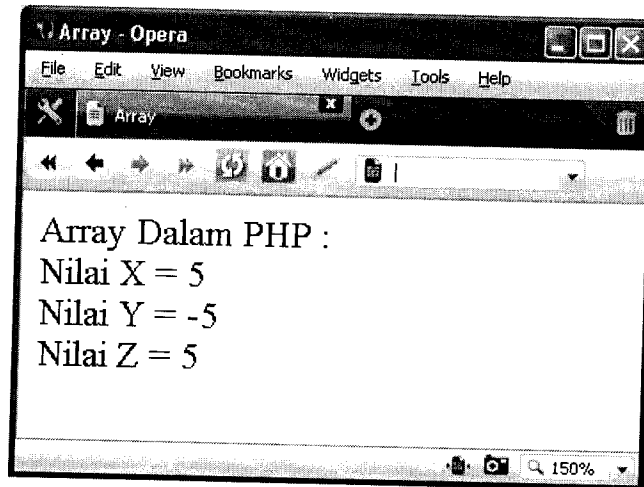
```
<HTML>
<HEAD>
<TITLE>Belajar PHP</TITLE>
</HEAD>
<BODY>
```

Contoh Penggunaan Array dalam PHP :


```

$z =-$y;
echo "Nilai X = $x <br>";
echo "Nilai Y = $y <br>";
echo "Nilai Z = $z <br>";
?>
</BODY>
</HTML>

```



Gambar 3.15 Array Numeric X, Y, Z

Contoh program:

```

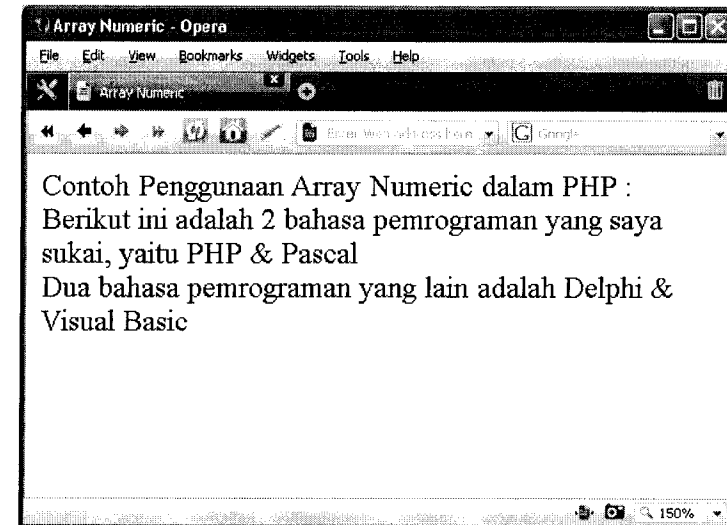
<HTML>
<HEAD>
<TITLE>Array Numeric</TITLE>
</HEAD>
<BODY>
Contoh Penggunaan Array Numeric dalam PHP : <BR>
<?
$pemrograman[0] = "PHP";

```

```

$pemrograman[1] = "Pascal";
$pemrograman[2] = "Delphi";
$pemrograman[3] = "Visual Basic";
echo "Berikut ini adalah 2 bahasa pemrograman yang saya sukai, yaitu " .
    $pemrograman[0] . " & " . $pemrograman[1];
echo "<br>Dua bahasa pemrograman yang lain adalah " . $pemrograman[2] .
    " & " . $pemrograman[3];
?>
</BODY>
</HTML>

```



Gambar 3.16 Array Numeric

Sintaks array untuk menyatakan suatu array berindeks numerik beserta nilainya adalah:

```
$nama_array[$key] = value;
```

Dalam hal ini, **\$key** dapat berupa bilangan bulat, mulai dari: 0, 1, 2, ...

Contoh:

```
$pemrograman[0] = "PHP";
$pemrograman[1] = "Pascal";
$pemrograman[2] = "Delphi";
$pemrograman[3] = "Visual Basic";
```

Array Asosiatif

Untuk membuat array berindex asosiatif, Anda membutuhkan label atau nama masing-masing elemen. Contoh di atas akan dikembangkan agar dapat menyimpan data nomor telepon masing-masing teman, dengan gambaran visualnya sebagai berikut:

Joko	Riyanto	Ani
123	456	789

Pada tabel tersebut, masing-masing nama akan dijadikan label untuk nilai nomor telepon yang disimpan. Kode pemrogramannya adalah sebagai berikut:

```
$telepon = array ("Joko"=>123, "Riyanto"=>456, "Ani"=>789);
```

atau

```
$telepon = array("Joko"=>123);
$telepon ["Riyanto"] = 456;
$telepon ["Ani"] = 789;
```

Dalam array asosiatif, kodenya sama dengan array yang berindeks numerik dan perbedaannya pada **\$key**. Pada array asosiatif, **\$key** dapat berupa suatu string. Perhatikan penggalan sintaks berikut:

```
$telepon["Joko"] = 123;
$telepon["Riyanto"] = 456;
$telepon["Ani"] = 789;
$telepon["Lisiani"] = 0;
```

Berdasar skrip tersebut, contoh kode untuk menampilkan nilai array asosiatif adalah sebagai berikut:

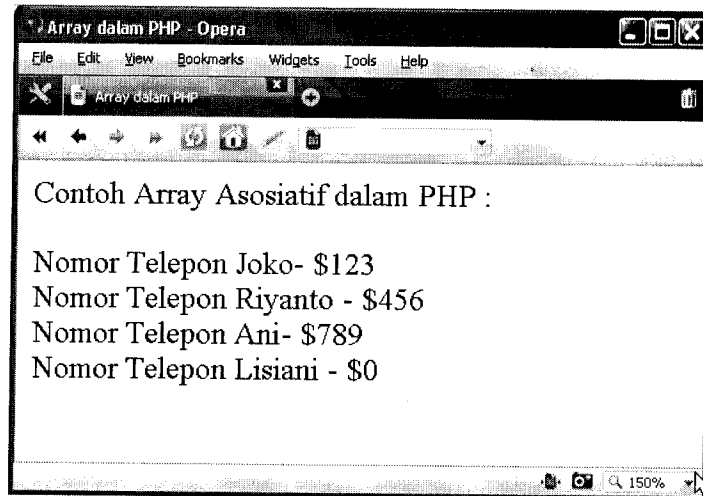
```
<HTML>
<HEAD>
<TITLE>Array dalam PHP</TITLE>
</HEAD>
<BODY>
```

Contoh Array Asosiatif dalam PHP :


```
<?
$telepon["Joko"] = 123;
$telepon["Riyanto"] = 456;
$telepon["Ani"] = 789;
$telepon["Lisiani"] = 0;

echo "Nomor Telepon Joko- $" . $telepon["Joko"] . "<br>";
echo "Nomor Telepon Riyanto-$" . $telepon["Riyanto"] . "<br>";
echo "Nomor Telepon Ani- $" . $telepon["Ani"] . "<br>";
echo "Nomor Telepon Lisiani-$" . $telepon["Lisiani"];

?>
</BODY>
</HTML>
```



Gambar 3.17 Array Asosiatif

Fungsi Array

Ada beberapa fungsi yang dapat digunakan dalam sebuah variabel array. Fungsi utama array adalah sebagai metode pengurutan data dengan perintah `sort`, `asort`, dan `ksort`.

Sort

Semua elemen pada array dapat diurutkan dengan menggunakan fungsi `sort()`. Contoh penggunaan skripnya adalah sebagai berikut:

```
sort($nama);
for($i=0; $i<length($nama);$i++){
echo("Array \$nama index ke [$i] adalah $nama [$i]");
echo("<br>");
}
```

Output yang dihasilkan:

```
Array nama index ke 0 adalah Joko
Array nama index ke 1 adalah Riyanto
Array nama index ke 2 adalah Ani
```

Asort dan Ksort

`Asort` digunakan untuk mengurutkan asosiatif array berdasarkan isinya. Contoh penggunaan:

```
asort($telepon);
while(list($index, $nilai) = each($telepon)){
echo("Array \$telepon index $index berisi nilai $nilai");
echo("<br>");
}
```

Output yang dihasilkan:

```
Array $ telepon index Joko berisi nilai 123
Array $ telepon index Riyanto berisi nilai 456
Array $ telepon index Ani berisi nilai 789
```

`Ksort` digunakan untuk mengurutkan asosiatif array berdasarkan index-nya. Contoh penggunaan:

```
ksort($telepon);
while(list($index, $nilai) = each($telepon)){
echo("Array \$ telepon index $index berisi nilai $nilai");
echo("<br>");
}
```

Output yang dihasilkan:

```
Array $ telepon index Ani berisi nilai 789
Array $ telepon index Riyanto berisi nilai 456
Array $ telepon index Joko berisi nilai 123
```

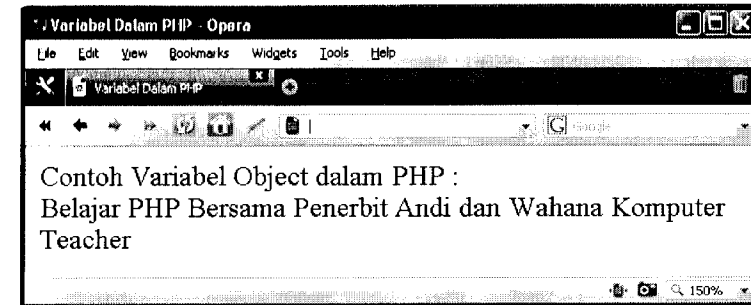
Objects

Object adalah sebuah tipe data yang dapat berupa sebuah bilangan, variabel, bahkan fungsi. Object merupakan instansiasi dari suatu *class*. Object dapat diinisialisasi menggunakan `statement new`. Object dibuat dengan maksud untuk membantu programmer yang telah terbiasa

dengan OOP (Object Oriented Programming), walaupun fasilitas OOP yang disediakan PHP masih sangat terbatas dalam PHP 4.

Perhatikan contoh berikut ini:

```
<HTML>
<HEAD>
<TITLE>Variabel Dalam PHP</TITLE>
</HEAD>
<BODY>
Contoh Variabel Object dalam PHP : <BR>
<?php
class data
{
var $nama ="Teacher";
function set_var($nama)
{
$this->nama = $nama;
}
}
$class = new data;
echo $class->nama;
$class->set var("Belajar tidak mengenal waktu dan usia Belajar PHP Bersama
Andi Offset dan Wahana Komputer");
echo ("<br>$class-> nama");
?>
</BODY>
</HTML>
```



Gambar 3.18 Variabel Konstanta

Dalam program di atas terdapat sebuah class dengan nama data, kemudian dibuat sebuah data object data dari class data, data yang di cetak dan keluar adalah **Teacher**. Class data mempunyai sebuah metode dengan nama **set_var**. Metode ini digunakan untuk memberi nilai pada variabel **\$nama**. Data object data yang dibuat dari class data akan mempunyai semua properties dari class data, termasuk method-nya, sehingga sebuah data bertipe object juga dapat mengandung sebuah method (fungsi). Pada program di atas, method **set_var** pada object data digunakan untuk mengubah nilai variabel **\$nama** menjadi **Teacher**, kemudian mencetaknya.



Karena PHP sangat lemah dalam hal struktur variabelnya, suatu array dapat dikombinasikan setiap menggunakan angka atau string. Aturan yang sangat penting dalam penggunaan array yaitu: tiap kunci array harus berlainan atau unik.

3.3.2 KONVERSI TIPE DATA

Tipe data tidak ditentukan secara eksplisit dalam pemrograman PHP karena tipe data ditetapkan sesuai konteksnya pada saat digunakan. Sebagai contoh, sebuah variabel **\$var** dideklarasikan bertipe integer, kemudian akan berubah otomatis menjadi bertipe *float* jika diberi nilai bilangan pecahan. Dalam beberapa kebutuhan, terkadang diperlukan konversi nilai dari satu tipe data ke tipe data yang lain. PHP menyediakan perintah untuk melakukan konversi tipe data dengan sintaks sebagai berikut:

```
Settype(value, datatype)
```

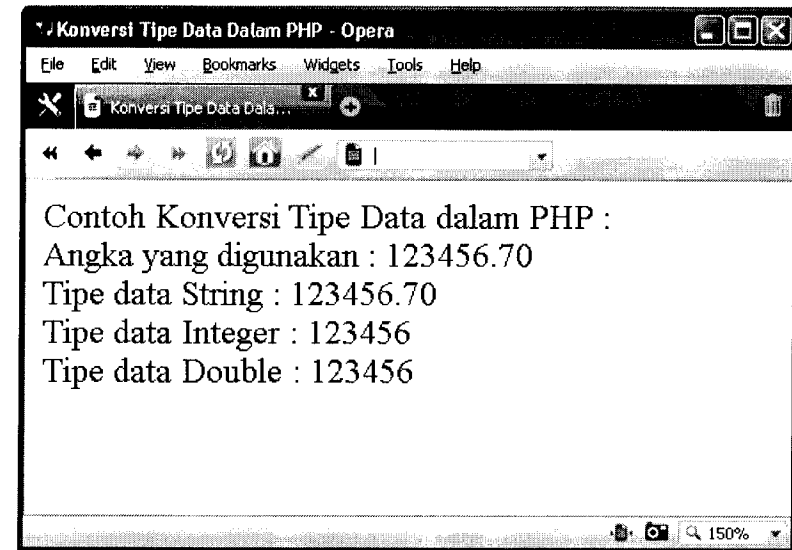
Keterangan:

- **Value:** nilai yang akan dikonversi.
- **datatype:** tipe data yang akan dikonversikan, misalnya bernilai "integer" atau "int", "string", "boolean", dan lain-lain.

Perhatikan contoh di bawah ini:

```
<HTML>
<HEAD>
<TITLE>Konversi Tipe Data Dalam PHP</TITLE>
</HEAD>
<BODY>
Contoh Konversi Tipe Data dalam PHP : <BR>
<?php
print("Angka yang digunakan : 123456.70 <BR>\n");
$Angka="123456.70";
// Penggunaan tipe data integer
print("Tipe data String : $Angka<BR> \n");
// Penggunaan tipe data string
settype($Angka, "Integer");
print("Tipe data Integer : $Angka<BR> \n");

//Penggunaan tipe data double
settype($Angka, "Double");
print("Tipe data Double : $Angka<BR> \n");
?>
</BODY>
</HTML>
```



Gambar 3.19 Konversi Data

Perhatikan contoh program pada **Gambar 3.19**. Pada umumnya, PHP juga mengenal adanya konversi data, seperti halnya bahasa pemrograman lain. Pada skrip tersebut, masing-masing tipe data memiliki perbedaan dalam hal range data yang dapat dibaca.

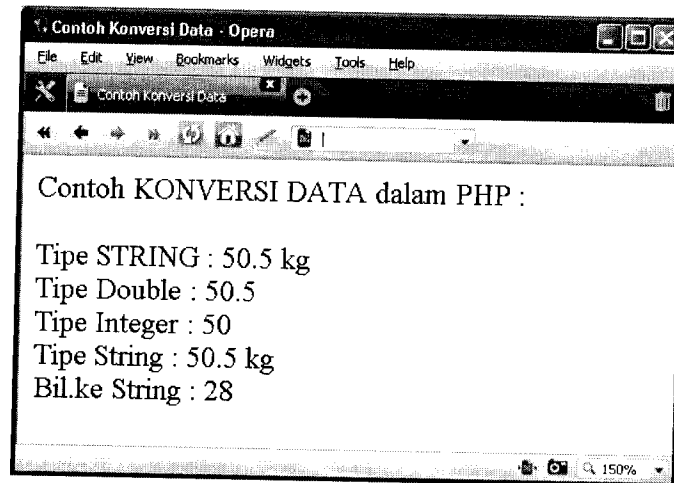
Contoh program:

```
<HTML>
<HEAD>
<TITLE>Contoh Konversi Data</TITLE>
</HEAD>
<BODY>
Contoh KONVERSI DATA dalam PHP : <BR> <BR>
<?php
$berat = "50.5 kg";
print("Tipe STRING : $berat <BR>\n");
printf("Tipe Double : %s <BR>\n", (double) $berat);
printf("Tipe Integer : %s <BR>\n", (int) ($berat));
```

```

printf("Tipe String : %s <BR>\n", $berat);
$bilangan = 28;
$berat = (string) $bilangan; // konversi ke string
printf("Bil.ke String : %s <BR>\n", $berat);
?>
</BODY>
</HTML>

```



Gambar 3.20 Konversi Data STRING

Contoh program sebelumnya juga menunjukkan bahwa konversi data juga dapat berfungsi sebagai pembanding dengan tipe data yang lain. Dengan menggunakan penanda “%s” maka skrip tersebut merupakan tipe data STRING.

3.4 VARIABEL

Variabel digunakan untuk menyimpan suatu nilai atau data sementara, seperti: text, angka, atau array. Ketika sebuah variabel dibuat, variabel tersebut dapat dipakai berulang-ulang, namun data yang disimpan dalam variabel akan hilang setelah program selesai dieksekusi.

Semua variabel yang terdapat pada PHP harus dimulai dengan karakter ‘\$’. Variabel PHP tidak perlu dideklarasikan dan ditetapkan jenis datanya

sebelum Anda menggunakan variabel tersebut. Hal tersebut menunjukkan bahwa tipe data dari variabel dapat berubah sesuai dengan perubahan konteks yang dilakukan oleh user. Secara tipikal, variabel PHP cukup diinisialisasikan dengan memberikan nilai kepada variabel tersebut. Panjang variabel tidak terbatas, setelah diawali ‘\$’ oleh huruf atau *underscore* (*_*), karakter berikutnya bisa terdiri dari huruf, angka, dan karakter tertentu yang diperbolehkan (karakter ASCII dari 127-255). Variabel bersifat *case-sensitive*, tidak perlu dideklarasikan, dan tidak boleh mengandung spasi.

Contoh penulisan nama variabel yang benar:

- `$_nama`
- `$nama_teacher`
- `$nama teacher`
- `$nama_ teacher`

Contoh penulisan nama variabel yang salah:

- `$5nama`
- `$nama?`
- `$nama+ teacher`
- `$nama. teacher`
- `$nama teacher`

Contoh berikut akan mencetak “PHP”:

```

$text = "PHP";
print "$text";

```

Identifier variabel dalam PHP tersebut bersifat *case-sensitive*, sehingga `$text` dan `$Text` merupakan variabel yang berbeda. *Built-in function* dan *structure* tidak *case-sensitive*, sehingga `echo` dan `ECHO` akan mengerjakan perintah yang sama. Identifier dapat berupa sejumlah huruf, digit/angka, *underscore*, atau tanda dollar, tetapi tidak dapat dimulai dengan digit/angka.

Contoh program:

```

<HTML>
<HEAD>
<TITLE>Variabel dalam PHP</TITLE>
</HEAD>

```

<BODY>

Contoh Variabel dalam PHP :

<?php

\$a = 4;

\$b = "Selamat Belajar PHP";

\$c = 14.2278;

\$d = 6;

echo "Variabel a = \$a
";

echo "\$b
";

echo "Variabel a + d = \$a + \$d
";

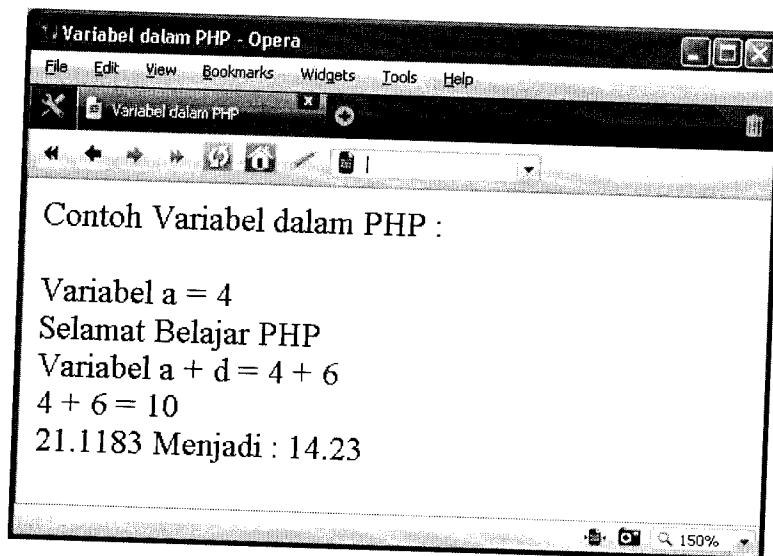
printf("%d + %d = %d
", \$a, \$d, \$a+\$d);

printf("21.1183 Menjadi :%4.2f", \$c);

?>

</BODY>

</HTML>



Gambar 3.21 Variabel dalam PHP

Contoh program:

<HTML>

<HEAD>

<TITLE>Variabel dalam PHP</TITLE>

</HEAD>

<BODY>

Contoh Variabel Konstanta dalam PHP :

<?php

\$Slogan="Belajar tidak mengenal waktu dan usia";

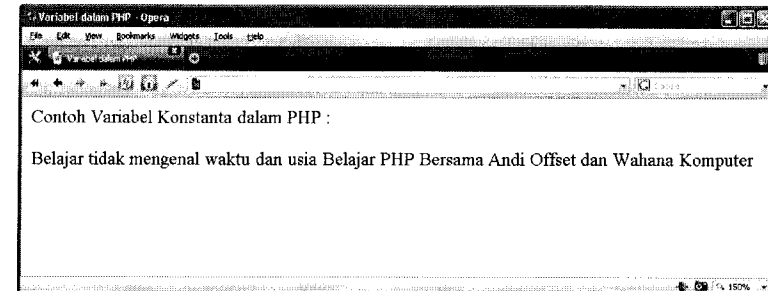
\$\$Slogan="Belajar PHP Bersama Andi Offset dan Wahana Komputer";

echo "\$Slogan \${\$Slogan}";

?>

</BODY>

</HTML>



Gambar 3.22 Variabel Konstanta

Dalam sintaks di atas, antarvariabel dapat digabungkan dengan dideklarasikan terlebih dahulu.

\$Slogan="Belajar tidak mengenal waktu dan usia";

\$\$Slogan="Belajar PHP Bersama Andi Offset dan Wahana Komputer";

Masing-masing variabel telah memiliki nilai. Untuk menggabungkan kedua variabel tersebut dalam PHP dapat dilakukan:

echo "\$Slogan \${\$Slogan}";

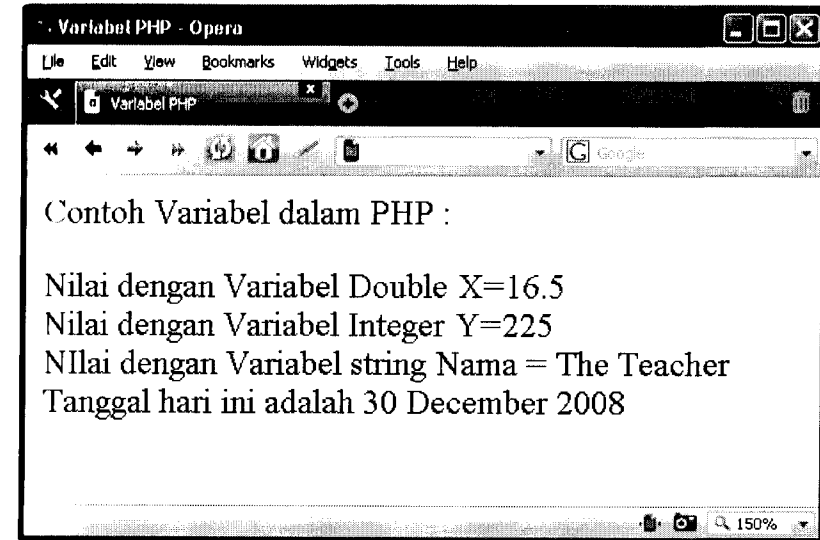
Variabel akan ditampilkan ke dalam server pada skrip *echo* tersebut.

Contoh program:

```
<HTML>
<HEAD>
<TITLE>Variabel PHP</TITLE>
</HEAD>
<BODY>
Contoh Variabel dalam PHP : <BR>
<?php
//variabel bertipe double
$X=6+10.5;
echo("Nilai dengan Variabel Double X=$X<BR>");

//variabel bertipe integer
$Y=225;
echo("Nilai dengan Variabel Integer Y=$Y<BR>");

//variabel bertipe string
$Nama="The Teacher";
echo("Nilai dengan Variabel string Nama = $Nama<BR>");
//Variabel bertipe date
$Tanggal=date("d F Y");
echo("Tanggal hari ini adalah $Tanggal<BR>");
?>
</BODY>
</HTML>
```



Gambar 3.23 Variabel menggunakan Tipe Data

Penjelasan skrip di atas merupakan penggunaan tipe variabel, yang pada dasarnya hampir mirip dengan tipe data. Pemanggilan nilai dalam variabel memiliki beberapa ketentuan, di antaranya:

```
//variabel bertipe double
$X=6+10.5;
echo("Nilai dengan Variabel Double X=$X<BR>");
```

Pada penggalan skrip tersebut, browser akan menampilkan hasil, yaitu: 16.5 karena bertipe DOUBLE.

```
//variabel bertipe integer
$Y=225;
echo("Nilai dengan Variabel Integer Y=$Y<BR>");
```

Integer variabel sama dengan penggunaan tipe data; integer memiliki jangkauan tersendiri.

```
//variabel bertipe string
$Nama="The Teacher";
echo("Nilai dengan Variabel string Nama = $Nama<BR>");
```

Variabel bertipe STRING dapat menampilkan nilai berupa karakter.

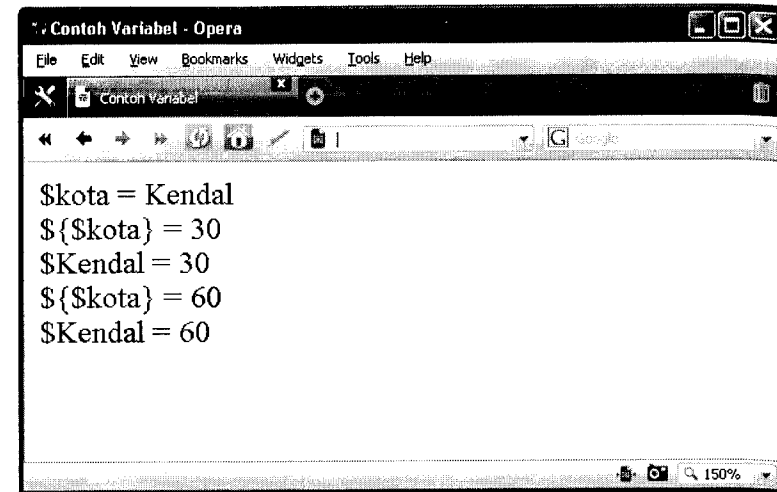
```
//Variabel bertipe date
$Tanggal=date("d F Y");
echo("Tanggal hari ini adalah $Tanggal<BR>");
```

Anda dapat menampilkan penanggalan hari ini (misalnya) dengan menggunakan variabel dengan tipe DATE.

Berdasarkan penjelasan tersebut, kita dapat mengambil kesimpulan bahwa variabel dalam PHP mempunyai berbagai macam tipe.

Contoh program:

```
<HTML>
<HEAD>
<TITLE>Contoh Variabel</TITLE>
</HEAD>
<BODY>
<?php
    $kota = "Kendal";
    ${$kota} = 30;
    print("\$kota = $kota <BR>\n");
    print("${$kota} = ${$kota} <BR>\n");
    print("\$Kendal = $Kendal <BR>\n");
    $Kendal = 60;
    print("${$kota} = ${$kota} <BR>\n");
    print("\$Kendal = $Kendal <BR>\n");
?>
</BODY>
</HTML>
```



Gambar 3.24 Variabel menggunakan Tipe Data

Contoh program pada Gambar 3.24 memiliki 2 (dua) bentuk variabel, yaitu:

\$kota = "Kendal";

\${\$kota} = 30;

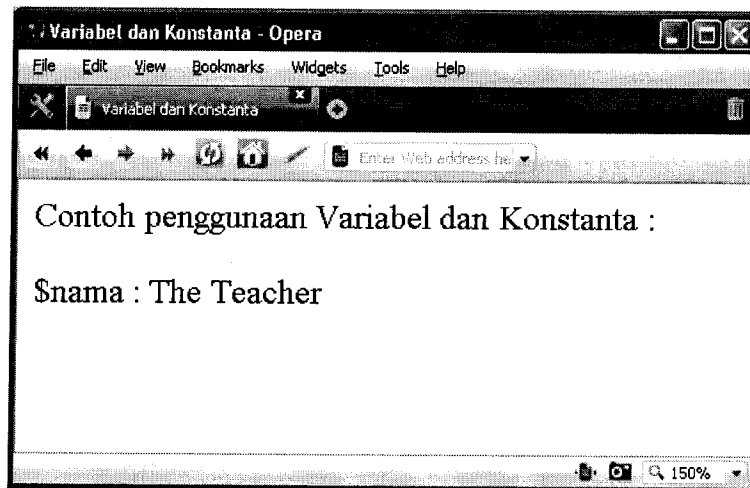
Variabel **\$kota** mempunyai nilai berupa kata "Kendal", sedangkan variabel **\${\$kota}** memiliki nilai 6000. Berdasarkan variabel tersebut, kita dapat menampilkan masing-masing nilai variabel:

```
$kota = "Kendal";
${$kota} = 30;
print("\$kota = $kota <BR>\n");
print("${$kota} = ${$kota} <BR>\n");
print("\$Kendal = $Kendal <BR>\n");
$Kendal = 60;
print("${$kota} = ${$kota} <BR>\n");
print("\$Kendal = $Kendal <BR>\n");
```

Skrip yang dicetak tebal menunjukkan bahwa nilai tersebut akan ditampilkan pada browser.

Contoh program:

```
<HTML>
<HEAD>
<TITLE>Variabel dan Konstanta</TITLE>
</HEAD>
<BODY>
Contoh penggunaan Variabel dan Konstanta : <BR> <BR>
<?php
    $nama = "The Teacher";
    echo("\$nama : $nama");
?>
</BODY>
</HTML>
```



Gambar 3.25 Variabel dengan Menggunakan STRING

Contoh program pada Gambar 3.25 menampilkan penggunaan variabel secara sederhana menggunakan variabel `$nama` dengan nilai berupa "The Teacher". Dalam kasus ini, nama variabel juga ikut ditampilkan dalam browser.

3.4.1 KONSTANTA

Konstanta merupakan variabel konstan yang nilainya tidak berubah. Anda dapat menggunakan fungsi `define()` untuk mendefinisikan konstanta dalam PHP karena konstanta merupakan variabel yang nilainya tetap. Konstanta hanya diberi nilai pada awal program dan nilainya tidak pernah berubah selama program berjalan.

PHP juga telah mendefinisikan beberapa konstanta, misalnya `PHP_VERSION`, yaitu konstanta yang memberikan informasi tentang versi PHP yang digunakan. Contoh lain, konstanta `TRUE` telah diberi nilai 1 dan `FALSE` dengan nilai 0.

Sintaks yang dapat digunakan untuk membuat konstanta adalah:

```
Define ("nama_konstanta","nilai konstanta");
```

Konstanta dideklarasikan hanya satu kali dan nilai tidak dapat diubah atau didefinisikan lagi pada keseluruhan program.

3.4.2 MENDEFINISIKAN KONSTANTA

Anda dapat mendefinisikan suatu konstanta dalam variabel dengan menggunakan perintah berikut:

Sintaks:

Define()

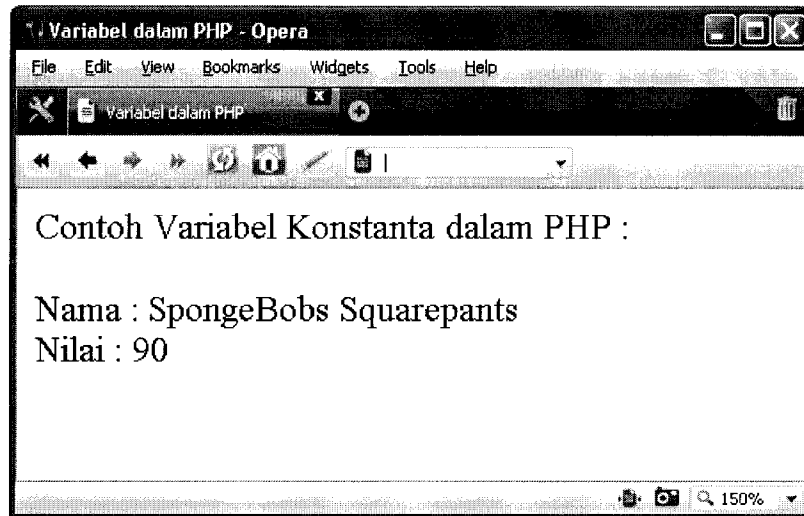
Contoh:

```
<HTML>
<HEAD>
<TITLE>Variabel dalam PHP</TITLE>
</HEAD>
<BODY>
```

Contoh Variabel Konstanta dalam PHP :


```
<?php
define ("NAMA", "SpongeBobs Squarepants");
define ("NILAI", 90);
```

```
Echo "Nama : " . NAMA;
Echo "<br>Nilai : " . NILAI;
?>
</BODY>
</HTML>
```



Gambar 3.26 Variabel Konstanta

Skrip tersebut merupakan penggunaan variabel konstanta dengan menggunakan atribut `define()`, sama halnya dengan variabel sebelumnya. Perbedaan variabel konstanta terletak pada fungsi `define`, yaitu `define("nama_konstanta,"nilai konstanta")`;; seperti yang terlihat pada penggalan skrip berikut:

```
define ("NAMA", "SpongeBobs Squarepants");
define ("NILAI", 90);
```

fungsi `define()` menggunakan `nama_konstanta` dan nilai konstanta adalah:

```
define ("NAMA", "SpongeBobs Squarepants");
```

"NAMA" adalah nama_konstanta sedang nilai_konstanta "SpongeBobs Squarepants".

```
define ("NILAI", 90);
```

"Nilai" adalah nama_konstanta sedang nilai_konstanta "90".

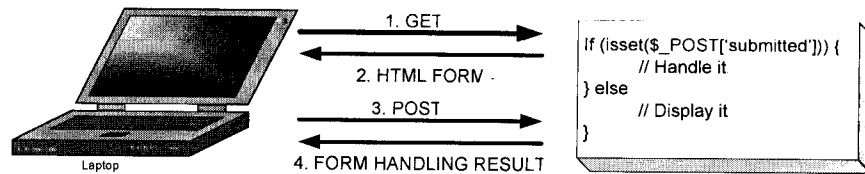
3.4.3 VARIABEL PHP

PHP ternyata memiliki variabel sendiri, tidak seperti bahasa pemrograman yang lain. Variabel `$HTTP_*_VARS` muncul jika konfigurasi `tracks_vars` aktif. Variabel PHP meliputi:

- `$HTTP_POST_VARS`: Hubungan array pada variabel yang dilayani oleh skrip melalui metode POST.
- `$HTTP_POST_FILES`: Hubungan array pada variabel yang berisi informasi tentang file yang di-upload melalui metode HTTP POST. `$HTTP_POST_FILES` tersedia mulai versi PHP 4 dan selanjutnya.
- `$HTTP_GET_VARS`: Hubungan array pada variabel yang dilayani oleh skrip melalui metode GET.
- `HTTP_ENV_VARS`: Hubungan array pada variabel yang dilayani oleh skrip melalui *parent environment*.
- `$HTTP_SERVER_VARS`: Hubungan array akan dilayani oleh skrip melalui HTTP Server.
- `$PHP_SELF`: Filename skrip yang sedang dieksekusi, relatif pada dokumen *root*.
- `$HTTP_COOKIE_VARS`: Hubungan array pada variabel yang dilayani oleh skrip melalui HTTP Cookies.

3.4.4 VARIABEL MELALUI HTML (GET DAN POST)

Ketika terdapat suatu form dalam skrip HTML, kemudian di-submit ke dalam skrip PHP, maka yang digunakan adalah bentuk variabel yang terdapat dalam PHP. Jika konfigurasi `track_vars` aktif, maka variabel ini akan mempunyai tempat dan merupakan hubungan array `$HTTP_POST_FILES`, `$HTTP_GET_VARS` dan `$HTTP_POST_VARS`.



Gambar 3.27 Struktur Akses Pengiriman Data GET dan POST

Pada PHP, Anda akan dikenalkan dengan bentuk variabel yang bertujuan untuk mengirimkan data dari dokumen HTML (GET dan POST) ke server, kemudian akan dieksekusi melalui file PHP setelah data terkirim (*submitted*). Jika bernilai **TRUE**, maka akan diolah menjadi file HTML untuk kemudian ditampilkan pada browser. Jika bernilai **FALSE**, maka pesan error akan muncul pada browser.

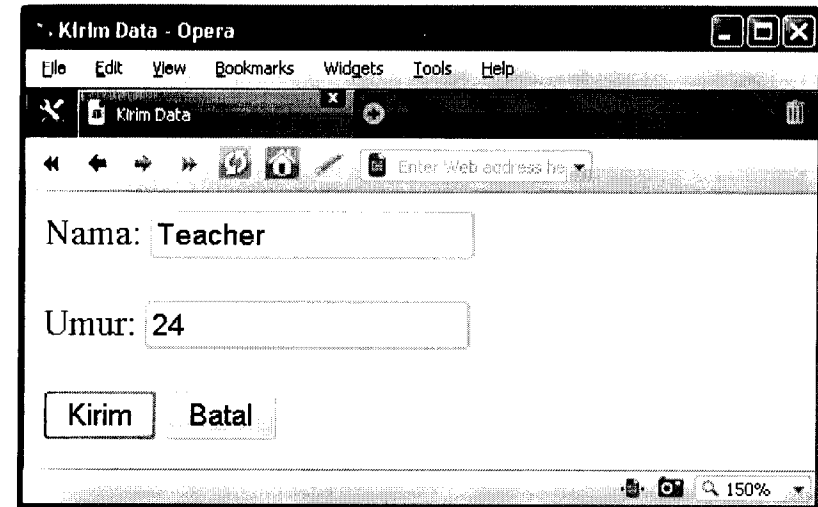
Contoh program:

```

<HTML>
<HEAD>
<TITLE>Kirim Data</TITLE>
</HEAD>
<BODY>
<form action="post.php" method="post">
Nama: <input type="text" name="name"><br><br>
Umur: <input type="text" name="umur"><br><br>
<input type="submit" Value="Kirim"> <input type="reset" Value="Batal">
</form>
</BODY>
</HTML>

```

Simpan dengan nama **Kirim_post.html**



Gambar 3.28 Form Input

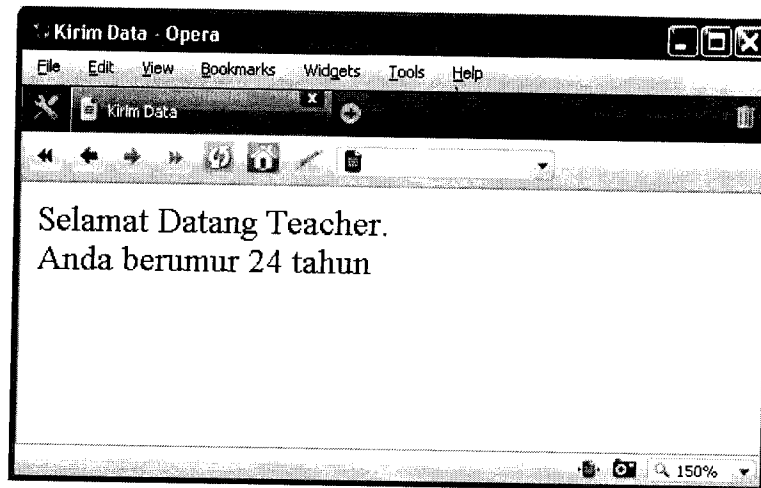
Selanjutnya, buatlah skrip PHP berikut dan simpan dengan nama **post.php**.

```

<HTML>
<HEAD>
<TITLE>Kirim Data</TITLE>
</HEAD>
<BODY>
Selamat Datang <?php echo $_POST["name"]; ?>.<br />
Anda berumur <?php echo $_POST["umur"]; ?> tahun
</BODY>
</HTML>

```

Setelah disimpan, buka skrip **Kirim_post.html**, isikan nama dan umur Anda, kemudian klik tombol **Kirim** hingga muncul tampilan seperti pada **Gambar 3.29**:

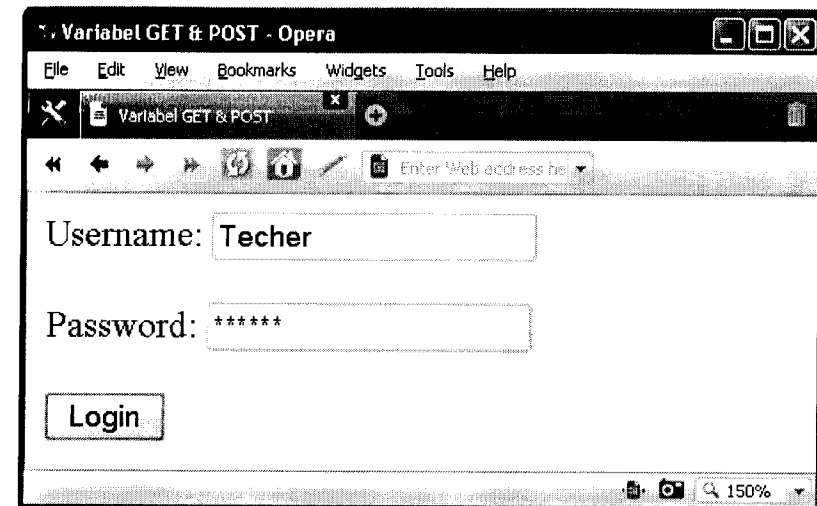


Gambar 3.29 Penggunaan Variabel untuk Menampilkan Input

Contoh program:

```
<HTML>
<HEAD>
<TITLE>Variabel GET & POST</TITLE>
</HEAD>
<BODY>
<FORM METHOD=GET ACTION=POST.php>
Username: <INPUT TYPE=text NAME=nama><BR>
Password: <INPUT TYPE=password NAME=password><BR>
<INPUT TYPE="submit" VALUE="Login">
</FORM>
</BODY>
</HTML>
```

Simpan dengan nama **Login.html**.



Gambar 3.30 Form Login

```
<HTML>
<HEAD>
<TITLE>Struktur GET & POST</TITLE>
</HEAD>
<BODY>
<?php
if($nama=="Teacher" && $password=="adadeh") //Case Sensitif
echo("Anda telah berhasil login");
else
echo("Username atau Password Anda masih salah");
?>
</BODY>
</HTML>
```

Simpan dengan nama **POST.php**.



Gambar 3.31 Konfirmasi Login

Jika username dan password yang dimasukkan sesuai dengan yang terdapat pada `proses.php`, maka pada browser akan menampilkan seperti yang terlihat pada **Gambar 3.31**, sehingga address bar akan menampilkan username dan password yang telah dimasukkan.



Gambar 3.32 Username dan Pasword pada Address Bar



Dalam variabel GET dan POST, penulisan variabel dan nilainya bersifat *case-sensitive*; perhatikan penulisan huruf besar dan kecil.

3.5 RANGKUMAN

Anda diharapkan agar mematuhi konsep dasar PHP dalam penulisan program, termasuk segala ketentuannya. Semua elemen yang terdapat pada PHP memiliki bentuk sintaks yang hampir mirip dengan C, dengan

beberapa penambahan yang diaplikasikan secara khusus pada website. Struktur dasar pemrograman PHP yang *embedded script* dengan penambahan tanda `<?>` yang dapat diintegrasikan ke dalam HTML merupakan solusi keamanan ketika *source code* hanya menampilkan tag HTML, sedangkan skrip PHP akan berjalan pada *server side*.

3.6 KUIS

1. Sebutkan jenis-jenis penggunaan komentar pada PHP dan jelaskan!
2. Ada berapa tipe data yang digunakan dalam PHP? Sebutkan!
3. Tipe data apakah yang hanya memiliki nilai TRUE dan FALSE?
4. Sebutkan beberapa peraturan yang harus diperhatikan pada array!
5. Perhatikan contoh penulisan variabel berikut:
 - \$5nama
 - \$nama?
 - \$nama+teacher
 - \$nama. teacher
 - \$nama teacher

Dari kelima contoh tersebut, buatlah aturan penulisan yang benar!

3.7 LATIHAN

Buatlah skrip PHP dengan menggunakan variabel GET dan POST, sesuai dengan ketentuan berikut:

- Input berupa:
 - Nama
 - Umur
 - komentar
- Beri keterangan bahwa "Data yang Anda masukkan telah terkirim !!!"
- Tampilkan data yang telah dimasukkan tersebut!

Sesi IV

Konstanta, Operator, Ekspresi

Durasi	2 jam
Jenis	Teori dan Praktik
Media	PHP 5, Web Browser.
Tujuan	Setelah membaca sesi ini, siswa diharapkan dapat: ❖ Memahami Konstanta dalam PHP 5. ❖ Mengerti operator yang digunakan dalam PHP 5. ❖ Mengerti Ekspresi pada PHP 5.

Dalam setiap pemrograman, kita mengenal adanya aturan tertentu yang harus digunakan, mulai dari penempatan huruf besar, konstanta, variabel, hingga operator. Pada dasarnya, masing-masing bahasa pemrograman memiliki konstanta, operator, dan ekspresi. Namun demikian, yang membedakan dari ketiganya dalam hal ekspresi hanya penulisannya.

4.1 KONSTANTA

Konstanta merupakan variabel yang nilainya tetap dan suatu identifier untuk nilai yang sederhana. Konstanta hanya diberi nilai pada awal program dan nilainya tidak pernah berubah selama program berjalan. Konstanta juga tidak memerlukan sintaks "\$" untuk menyatakannya.

PHP memiliki fungsi **define()** untuk menyatakan suatu konstanta. Syarat penulisan konstanta adalah dapat diawali dengan menggunakan karakter, tidak boleh diawali dengan angka, dan dapat diawali dengan menggunakan tanda *underscore* (_).

Perbedaan antara variabel dengan konstanta adalah:

- Konstanta tidak diawali dengan tanda \$.
- Konstanta hanya mampu menerima data *scalar* (data yang hanya meliputi: integer string, boolean, dan double).
- Konstanta dapat digunakan dari tempat berbeda tanpa harus didefinisikan scope-nya.

Skrip untuk membuat konstanta:

```
Define ("nama_konstanta","nilai konstanta");
```

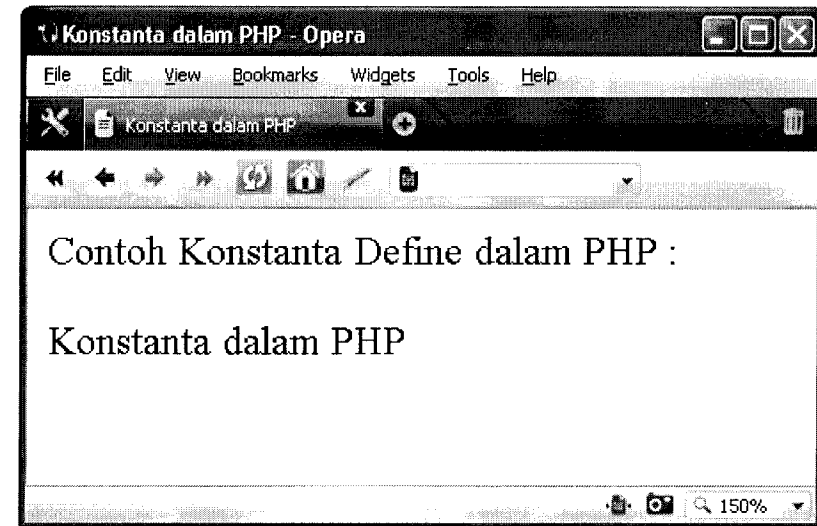
Konstanta hanya dideklarasikan satu kali dan nilai tersebut tidak dapat diubah atau didefinisikan lagi dalam seluruh program.

Contoh program:

```
<HTML>
<HEAD>
<TITLE>Konstanta dalam PHP</TITLE>
</HEAD>
<BODY>
```

Contoh Konstanta Define dalam PHP :


```
<?php
define ("Konstanta", "Konstanta dalam PHP");
echo (Konstanta);
?>
</BODY>
</HTML>
```



Gambar 4.1 Konstanta Define

Konstanta Define pada PHP sebenarnya hampir sama dengan penggunaan konstanta dalam variabel. Perbedaan hanya terletak pada aturan-aturannya.

Contoh program:

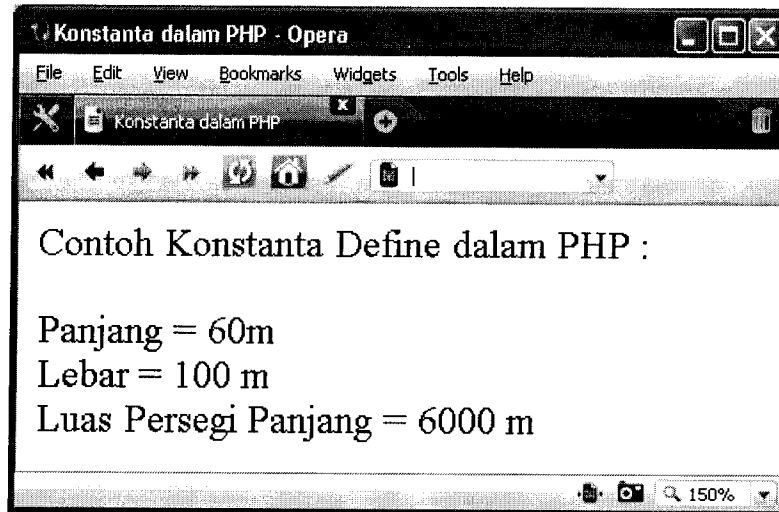
```
<HTML>
<HEAD>
<TITLE>Konstanta dalam PHP</TITLE>
</HEAD>
<BODY>
```

Contoh Konstanta Define dalam PHP :


```

<?php
define("x", 60);
$y = 100;
$luas=x*$y;
echo ("Panjang = ");
echo (x);
echo("m<br>");
echo ("Lebar = $y m<br>");
echo ("Luas Persegi Panjang = $luas m");
?>
</BODY>
</HTML>

```



Gambar 4.2 Konstanta Define dalam Mencari Luas Persegi Panjang

Selain mendefinisikan suatu nilai dalam suatu variabel, konstanta dalam PHP dapat juga digunakan dalam fungsi matematika. Penggunaan fungsi **define()** dimasukkan nilai yang sudah dideklarasikan, seperti:

```
define("x", 60);
```

Anda dapat menggunakan fungsi array untuk memanggil nilai yang sudah dideklarasikan. Dalam mencari Luas Persegi Panjang diketahui:

Nilai x = 60 m.

Nilai y = 100 m.

Rumus Panjang = Panjang x Lebar.

Panjang = x * y.

Contoh program:

```

<HTML>
<HEAD>
<TITLE>Konstanta dalam PHP</TITLE>
</HEAD>
<BODY>

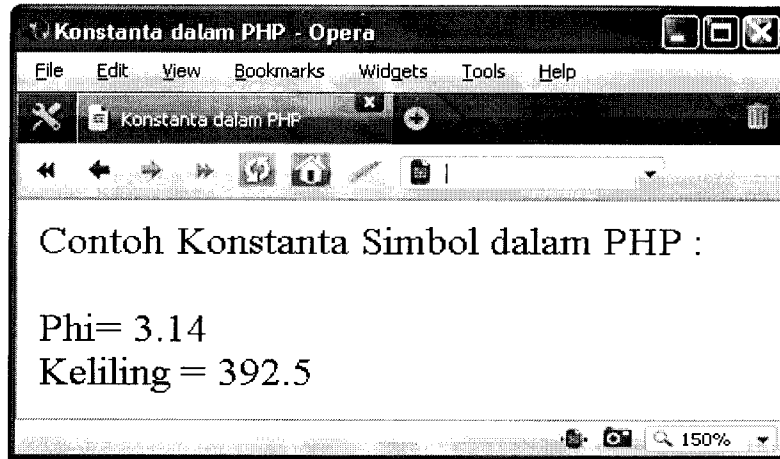
```

Contoh Konstanta Simbol dalam PHP :


```

<?php
define("phi", 3.14);
$jari_jari = 25;
$keliling = 5 *phi * $jari_jari;
printf("Phi= %s<BR>\n", phi);
printf("Keliling = %s <BR>\n", $keliling);
?>
</BODY>
</HTML>

```



Gambar 4.3 Konstanta Define Simbol Phi dalam Mencari Keliling Lingkaran

Sama halnya dengan contoh program sebelumnya, `define()` dimasukkan nilai yang bersifat konstan, yaitu *phi* = 3.14.

```
define("phi", 3.14);
```

Oleh karenanya, setelah nilai phi diketahui sebesar 3.14, maka gunakan fungsi matematika dalam PHP, seperti yang terlihat pada penggalan skrip berikut:

```
$keliling = 5 * phi * $jari_jari;
```

`$jari_jari` sama seperti fungsi array, yang jika dipanggil dapat digunakan setelah nilai `$jari_jari` dideklarasikan.

Contoh program:

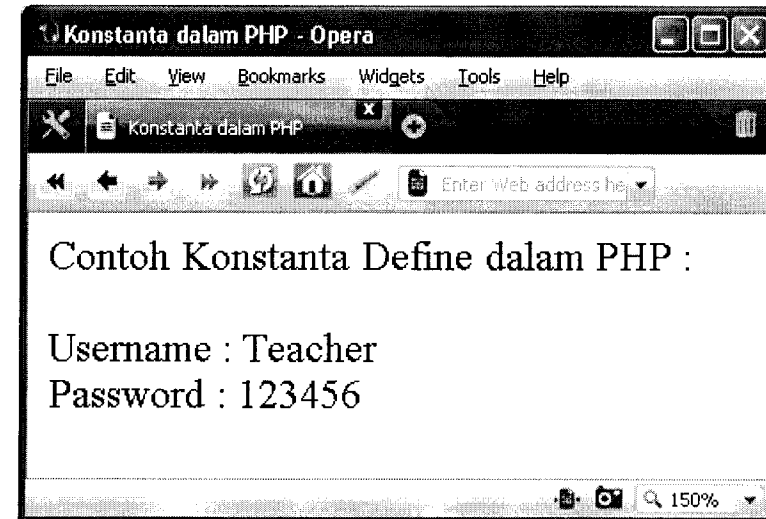
```
<HTML>
<HEAD>
<TITLE>Konstanta dalam PHP</TITLE>
</HEAD>
<BODY>
```

Contoh Konstanta Define dalam PHP :


```
<?php
```

```
define ("USERNAME", "Teacher");
```

```
define ("PASSWORD", 123456);
echo "Username : " . USERNAME;
echo "<br>Password : " . PASSWORD;
?>
</BODY>
</HTML>
```



Gambar 4.4 Konstanta Predefined dalam Penggunaan Akses Login

Berbeda halnya dengan contoh program sebelumnya, berikut ini merupakan contoh penggunaan dua buah fungsi `define()` dengan aturan yang sama.

```
define ("USERNAME", "Teacher");
define ("PASSWORD", 123456);
echo "Username : " . USERNAME;
echo "<br>Password : " . PASSWORD;
```

fungsi tersebut juga dapat digunakan sebagai validitas pengecekan username dan password yang akan dimasukkan.

4.1.1 KONSTANTA PREDEFINED

Pada PHP, konstanta yang banyak digunakan adalah:

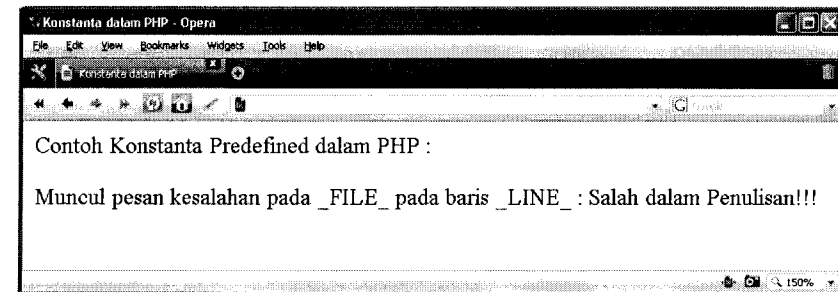
- **_FILE_**
File yang digunakan sebagai konstanta untuk skrip berfungsi mempresentasikan file yang sedang di-parse oleh PHP. Konstanta ini bersifat *case-sensitive*.
- **_LINE_**
Konstanta yang dapat menunjukkan nomor baris dari suatu skrip file yang sedang di-parse oleh PHP. Konstanta ini juga bersifat *case-sensitive*.
- **TRUE**
Value TRUE digunakan untuk tipe data Boolean; juga bersifat *case-sensitive*.
- **FALSE**
Value FALSE juga digunakan untuk tipe data Boolean; konstanta ini bersifat *case-sensitive*.
- **NULL**
Value NULL digunakan untuk tipe NULL dan bersifat *case-sensitive*.
- **PHP_OS**
Sistem operasi yang digunakan, parser PHP akan dieksekusi, seperti: MacOS, LINUX, SunOS, Win32, WINNT.
- **E_ERROR**
Pesan error saat terjadi *parsing error*.
- **E_WARNING**
Konstanta yang akan memberikan catatan pada saat PHP menemukan kesalahan.

Contoh program:

```
<HTML>
<HEAD>
<TITLE>Konstanta dalam PHP</TITLE>
</HEAD>
<BODY>
```

Contoh Konstanta Predefined dalam PHP :


```
<?php
function kesalahan($file, $baris, $pesan)
{
    echo "Muncul pesan kesalahan pada $file pada baris $baris : $pesan";
}
kesalahan(_FILE_, _LINE_, "Salah dalam Penulisan!!!");
?>
</BODY>
</HTML>
```



Gambar 4.5 Konstanta Predefined

4.2 OPERATOR

Operator adalah simbol yang digunakan untuk memanipulasi data, seperti penambahan dan pengurangan. Selain itu, operator juga digunakan untuk mengoperasikan *operand*, baik tunggal atau lebih dari satu. Ada operator yang menggunakan satu *operand* dan ada pula yang menggunakan dua *operand*.

Operator dapat dikelompokkan dalam empat macam, yaitu:

- **Operator aritmatika/arithmetic operator:** operator yang berhubungan dengan operasi matematika.
- **Operator logika/logical operator:** operator yang membandingkan kondisi dan menggabungkan sebuah ekspresi bernilai true dan false.

- **Operator perbandingan/comparison operator:** operator yang membandingkan *binary*.
- **Operator penugasan/assignment operator:** operator yang digunakan untuk memberikan nilai ke suatu variabel atau dari variabel ke variabel.

4.2.1 OPERATOR ARITMATIKA

Operator Aritmatika merupakan operator yang berhubungan dengan fungsi matematika, misalnya:

```
$a = 9 + 2;
```

Operator "+" berfungsi untuk menambahkan kedua *operand* (9 dan 2).

Ada beberapa operator aritmatika, yaitu:

- + : penjumlahan
- - : pengurangan
- * : perkalian
- / : pembagian
- % : nilai sisa pembagian

Tabel 4.1 Tabel Operator Aritmatika

Contoh	Nama	Hasil
\$a + \$b	Penjumlahan	Jumlah dari \$a dan \$b
\$a-\$b	Pengurangan	Pengurangan selisih dari \$a dan \$b
\$a * \$b	Perkalian	Perkalian hasil kali dari \$a dan \$b
\$a / \$b	Pembagian	Pembagian dari \$a dan \$b
\$a % \$b	Modulus	Sisa pembagian dari \$a dibagi \$b

Untuk lebih jelas, perhatikan Tabel 4.2:

Tabel 4.2 Contoh Penghitungan Aritmatika

Operator	Fungsi	Contoh	Penjelasan
+	Penjumlahan	5 + 6	Menghasilkan nilai : 11
-	Pengurangan	6-4	Menghasilkan nilai : 2
*	Perkalian	5 * 6	Menghasilkan nilai : 30
/	Pembagian	16 / 4	Menghasilkan nilai : 4
%	Modulus	7 % 43	Menghasilkan nilai : 1

PHP mengabaikan karakter spasi dalam sebuah operasi. Ekspresi `$x = 5 + 6` dengan `$x=5+6` akan menghasilkan nilai yang sama. Penggunaan spasi disesuaikan dengan keinginan programmer.

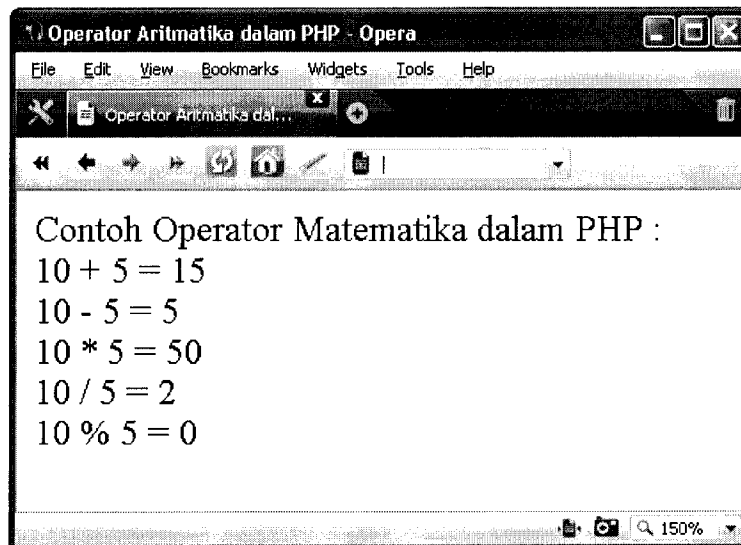
Contoh program:

```
<HTML>
<HEAD>
<TITLE>Operator Aritmatika dalam PHP</TITLE>
</HEAD>
<BODY>
Contoh Operator Matematika dalam PHP: <BR>
<?php
$a=10;
$b=5;
$c=$a+$b;
$d=$a-$b;
$e=$a*$b;
$f=$a/$b;
$g=$a%$b;
echo "$a + $b = $c <br>";
echo "$a-$b = $d <br>";
```

```

echo "$a * $b = $e <br>";
echo "$a / $b = $f <br>";
echo "$a % $b = $g <br>";
?>
</BODY>
</HTML>

```



Gambar 4.6 Operator Matematika Rumus Penghitungan

Tanpa menggunakan fungsi (array, fungsi, dan lain-lain) PHP dapat melakukan suatu proses. Sebagai contoh, melakukan penghitungan bilangan desimal dengan langkah awal deklarasi nilai yang akan digunakan.

```

$a=10;
$b=5;

```

Setelah mengetahui besar nilai yang akan digunakan, masukkan array yang akan digunakan:

```

$c=$a+$b;
$d=$a-$b;

```

```

$e=$a*$b;
$f=$a/$b;
$g=$a%$b;

```

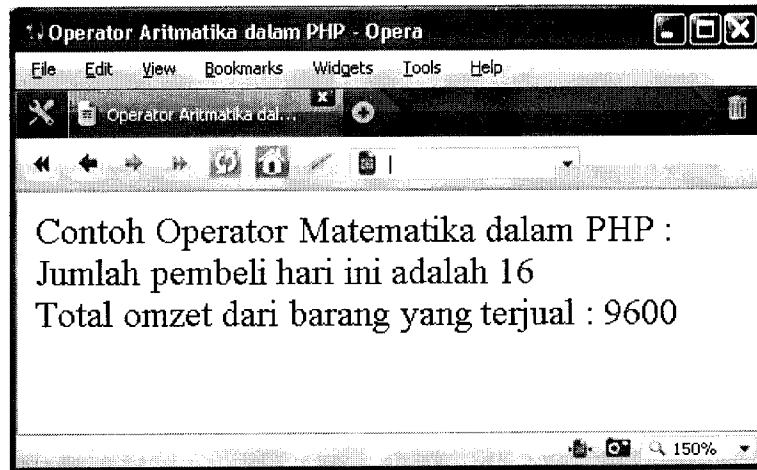
Secara otomatis, server akan menghitung nilai array yang telah dimasukkan hingga menampilkan hasil penghitungannya.

Contoh program:

```

<HTML>
<HEAD>
<TITLE>Operator Aritmatika dalam PHP</TITLE>
</HEAD>
<BODY>
Contoh Operator Matematika dalam PHP : <BR>
<?php
$harga=600;
$pembeli=16;
echo ("Jumlah pembeli hari ini adalah $pembeli <br>\n");
echo ("Total omzet dari barang yang terjual : ");
echo ($harga * $pembeli);
?>
</BODY>
</HTML>

```



Gambar 4.7 Operator Matematika Perkalian

4.2.2 OPERATOR LOGIKA

Operator Logika atau **Logical Operator** digunakan untuk membandingkan dua nilai variabel bertipe Boolean. Penggabungan dua nilai variabel biasanya akan menghasilkan sebuah ekspresi bernilai TRUE atau FALSE. Hasil yang didapatkan dari penggunaan *Logical Operator* adalah Boolean.

Tabel logika pada Tabel 4.3 digunakan sebagai pedoman perhitungan:

Tabel 4.3 Operator Logika

Operator	Contoh	Bernilai benar jika :
AND / and	\$x && \$y atau \$x AND \$y	\$x dan \$y bernilai benar
OR / or	\$x \$y atau \$x OR \$y	Salah satu atau kedua variabel bernilai benar
XOR	\$x XOR \$y	Salah satu variabel bernilai benar, tetapi tidak keduanya benar
NOT	!\$x	\$x tidak bernilai benar

Perhatikan Tabel 4.4 yang menunjukkan daftar prioritas penghitungan dalam operator logika:

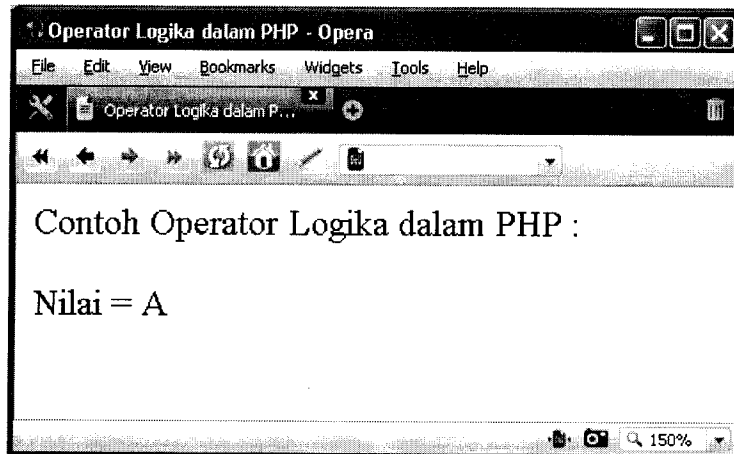
Tabel 4.4 Prioritas Penggunaan Operator

Prioritas	Operator
Tertinggi	(), {}
	!, ++, --, \$, &, ~
	?, %, *
	+, -
	<>, <=, >=
	&
	^
	!
	&&
	=, -=, +=, *=, /=, &=, =, ^=, .=
	AND (&&)
	XOR ()
Terendah	OR

Contoh program:

```
<HTML>
<HEAD>
<TITLE>Operator Logika dalam PHP</TITLE>
</HEAD>
<BODY>
Contoh Operator Logika dalam PHP : <BR> <BR>
<?php
$x=85;
if($x> 80 && $x <=100) echo "Nilai = A";
```

```
?>
</BODY>
</HTML>
```

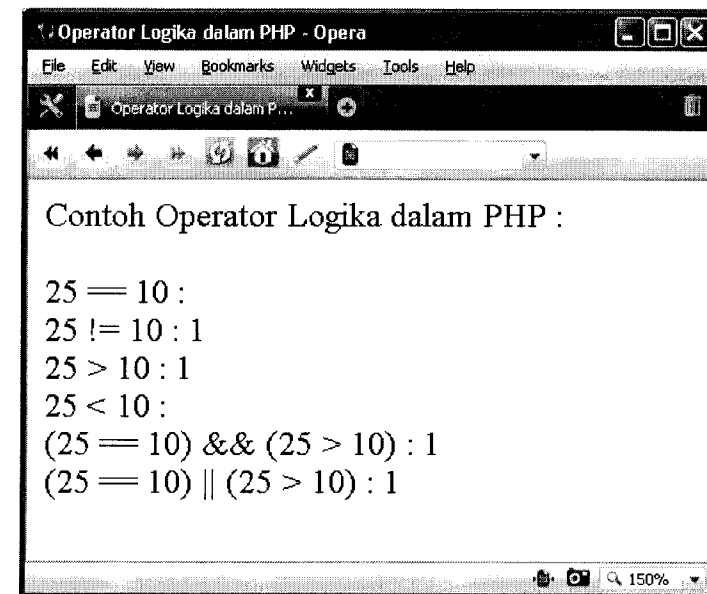


Gambar 4.8 Operator Logika Suatu Nilai

Contoh program:

```
<HTML>
<HEAD>
<TITLE>Operator Logika dalam PHP</TITLE>
</HEAD>
<BODY>
Contoh Operator Logika dalam PHP : <BR> <BR>
<?php
$x=25;
$y=10;
echo "$x == $y : ", ($x == $y);
echo "<br> $x != $y : ", ($x != $y);
echo "<br> $x > $y : ", ($x > $y);
echo "<br> $x < $y : ", ($x < $y);
```

```
echo "<br> ($x == $y) && ($x > $y) : ", (($x != $y) && ($x > $y));
echo "<br> ($x == $y) || ($x > $y) : ", (($x != $y) || ($x > $y));
?>
</BODY>
</HTML>
```



Gambar 4.9 Operator Logika Nilai X dan Y

Contoh program pada **Gambar 4.9** menunjukkan bahwa operator logika dapat digunakan untuk mencari perbandingan suatu bilangan, namun jika Anda tidak dapat menemukan hasil perbandingan, maka server tidak akan menampilkan hasil penghitungan.

4.2.3 OPERATOR PERBANDINGAN

Operator Perbandingan atau **Comparison Operator** adalah operator yang biasanya digunakan untuk melakukan perbandingan atau relasional antara dua buah *operand* dan menghasilkan nilai **true** atau **false**. Operator ini juga dapat digunakan untuk melakukan suatu pengujian kondisi.

Penggunaan tanda pembandingan (seperti ==, <, >, dan simbol lainnya) merupakan salah satu ciri dari operator perbandingan ini.

Perhatikan **Tabel 4.5** untuk mengenal beberapa simbol penggunaan operator perbandingan lainnya.

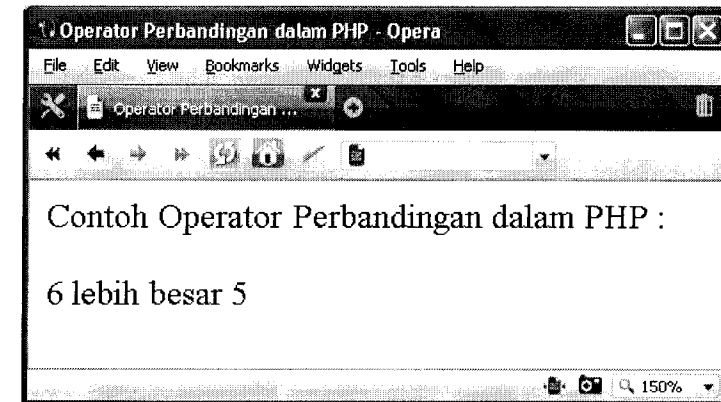
Tabel 4.5 Operator Perbandingan

Operator	Fungsi	Contoh	Menghasilkan benar (true) Jika :
==	Sama dengan	\$x == \$y	\$x dan \$y mempunyai nilai yang sama
<	Kurang dari	\$x < \$y	\$x kurang dari \$y
>	Lebih dari	\$x > \$y	\$x lebih dari \$y
<=	Kurang dari atau sama dengan	\$x <= \$y	\$x kurang dari atau sama dengan \$y
>=	Lebih dari atau sama dengan	\$x >= \$y	\$x lebih dari atau sama dengan \$y
!=	Tidak sama dengan	\$x != \$y	\$x tidak sama dengan \$y
<>	Tidak sama dengan	\$x <> \$y	\$x tidak sama dengan \$y
===	Identik	\$n === \$m	Benar jika \$n memiliki nilai yang sama dengan \$m, dan type data keduanya sama. (hanya dalam PHP4)

Contoh program:

```
<HTML>
<HEAD>
<TITLE>Operator Perbandingan dalam PHP</TITLE>
</HEAD>
<BODY>
Contoh Operator Perbandingan dalam PHP : <BR> <BR>
<?php
```

```
$n = 6;
if($n >=5) echo (" $n lebih besar 5 <br>");
?>
</BODY>
</HTML>
```



Gambar 4.10 Operator Perbandingan Mencari Perbandingan Besar Suatu Bilangan

Contoh program pada **Gambar 4.10** menunjukkan bahwa operator perbandingan menentukan nilai variabel yang akan digunakan untuk perbandingan. Penggunaan if merupakan suatu logika perbandingan bahwa nilai x > y atau sebaliknya.

4.2.4 OPERATOR PENUGASAN

Operator Penugasan atau Assignment Operator merupakan operator yang digunakan untuk memberikan nilai ke suatu variabel atau variabel ke variabel. Simbol yang biasa digunakan dalam operator ini adalah "sama dengan" (=). Perhatikan **Tabel 4.6**.

Tabel 4.6 Operator Penugasan

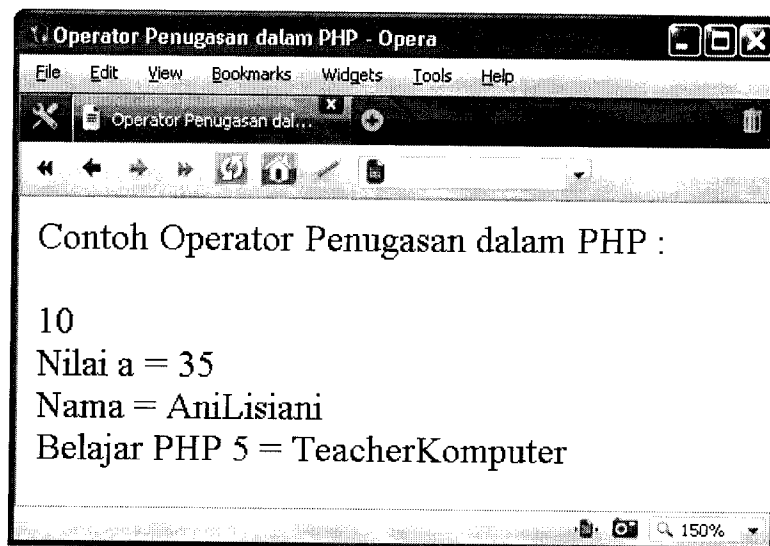
Contoh	Ekuivalen dengan	Fungsi	Contoh
$\$x += \y	$\$x = \$x + \$y$	Untuk menambah nilai variabel di sebelah kiri dengan nilai di sebelah kanan.	$X += 2;$ $\rightarrow x = x + 2;$
$\$x -= \y	$\$x = \$x - \$y$	Untuk mengurangi nilai variabel di kiri dengan nilai di sisi kanan.	$X -= 2;$ $\rightarrow x = x - 2;$
$\$x *= \y	$\$x = \$x * \$y$	Untuk melakukan perkalian antara variabel sebelah kiri dan sebelah kanan.	$X *= 2;$ $\rightarrow x = x * 2;$
$\$x /= \y	$\$x = \$x / \$y$	Untuk membagi nilai variabel di sisi kiri dengan nilai kanan.	$X /= 2;$ $\rightarrow x = x / 2;$
$\$x \% = \y	$\$x = \$x \% \$y$	Sisa hasil bagi antara nilai variabel kiri dengan nilai kanan.	$X \% = 2;$ $\rightarrow x = x \% 2;$
$\$x \&= \y	$\$x = \$x \& \$y$	Untuk melakukan operasi AND antara nilai variabel kiri dengan nilai di kanan.	$X \&= 2;$ $\rightarrow x = x \& 2;$
$\$x = \y	$\$x = \$x \$y$	Melakukan operasi OR antara nilai variabel kiri dan nilai variabel kanan.	$X = 2;$ $\rightarrow x = x 2;$
$\$x \wedge = \y	$\$x = \$x \wedge \$y$	Untuk melakukan operasi XOR antara nilai variabel kiri dengan nilai variabel kanan.	$X \wedge = 2;$ $\rightarrow x = x \wedge 2;$
$\$x .= \y	$\$x = \$x . \$y$	Untuk melakukan operasi gabungan antara nilai variabel di sisi kiri dengan nilai di sisi kanan.	$X .= "joko";$ $\rightarrow x = x . "joko";$

Contoh program:

```

<HTML>
<HEAD>
<TITLE>Operator Penugasan dalam PHP</TITLE>
</HEAD>
<BODY>
Contoh Operator Penugasan dalam PHP : <BR> <BR>
<?php
echo
    $a = 10;
    $a+=25;
echo "<br>";
    echo "Nilai a = $a <br>";
    $n = 'Ani';
    $n.='Lisiani';
    echo "Nama = $n <br>";
    $m = 'Teacher';
    $m = $m.'Komputer';
    echo "Belajar PHP 5 = $m <br>";
?>
</BODY>
</HTML>

```



Gambar 4.11 Operator Penugasan

Sebagaimana fungsinya, operator penugasan memberikan nilai ke dalam suatu variabel, yang kemudian akan digunakan untuk keperluan berbagai proses. Nilai yang sudah dimasukkan ke dalam suatu variabel dapat ditampilkan, dihitung, dan dibandingkan dengan nilai atau variabel yang lain.



Tanda (==) digunakan untuk operator perbandingan untuk menguji suatu variabel, sedangkan tanda (=) adalah operator penugasan untuk memberikan nilai kepada suatu variabel.

4.3 OPERATOR YANG LAIN

Selain jenis operator yang telah dijelaskan sebelumnya, PHP juga memiliki operator lainnya. Hal tersebut bukan karena jarang digunakan, tetapi memang fungsi dari variabel berikut memiliki tujuan khusus.

4.3.1 UNARY OPERATOR

Operator Unary digunakan secara tunggal pada suatu *operand*. Tanda minus (-) digunakan dengan sebuah nilai numerik tunggal untuk menegatifkan sebuah bilangan (untuk membuat negatif bilangan positif atau membuat positif bilangan negatif).

Contoh:

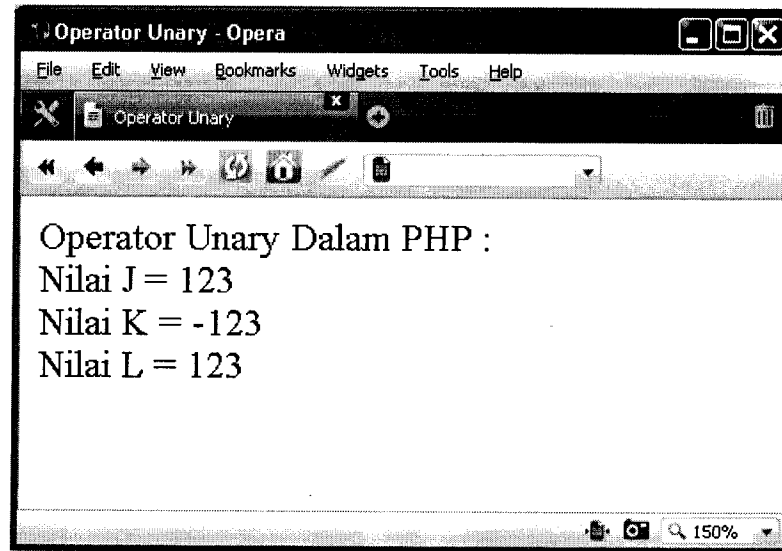
```
$x = 123;
$y = -$x; // $y = -123
$j = -456;
$k = -$j; // $k = 456
```

Contoh penggunaan Operator Unary dalam PHP adalah:

```
<HTML>
<HEAD>
<TITLE>Operator Unary</TITLE>
</HEAD>
<BODY>
Operator Unary Dalam PHP : <BR>
<?php
$J=123;
$K=-$J;
$L =-$K;

echo "Nilai J = $J <br>";
echo "Nilai K = $K <br>";
echo "Nilai L = $L <br>";
?>
</BODY>
</HTML>
```

Selanjutnya, Anda akan melihat munculnya tampilan seperti yang terlihat pada **Gambar 4.12** pada browser.



Gambar 4.12 Unary Operator

4.3.2 STRING CONCATENATION OPERATOR

Operator ini menggunakan tanda titik (.) sebagai operator *concatenate* untuk menggabungkan dua atau lebih nilai string menjadi sebuah string tunggal.

Contoh:

```
$subjek = "Belajar PHP";
$predikat = "Bersama Andi dan Wahana";
$objek = "PHP 5";
$kalimat = $subjek . " " . $predikat . " " . $objek;
print $kalimat; // akan mencetak Belajar PHP Bersama Andi dan Wahana PHP 5
print "$kalimatversi 5"; // akan akan mencetak 5
print "$kalimat versi 5"; // akan mencetak Belajar PHP Bersama Andi dan Wahana PHP versi 5
print "${kalimat} versi 5"; // akan mencetak Belajar PHP Bersama Andi dan Wahana PHP versi 5
```

```
$bilangan = "<B>A</B>&nbsp;";
$bilangan .= "<I>B</I>&nbsp;";
$bilangan .= "<U>C</U>";
print $bilangan; // akan mencetak A B C
```

4.3.3 OPERATOR BITWISE

Operator Bitwise digunakan untuk perbandingan tingkat rendah, termasuk untuk memanipulasi bilangan biner dan membandingkan nilai biner. Format penulisan operator ini sama dengan operator logika, yaitu dengan menggunakan NOT, AND, OR, atau XOR.

Tabel 4.7 Operator Bitwise

Operator	Arti	Contoh
&	AND	15 (biner 1111) & 12 (biner 1100) 7 (biner 0111)
	OR	15 (biner 1111) 12 (biner 1101) 11 (biner 1011)
^	Exclusive OR	15 (biner 1111) ^ 12 (biner 1101) 7 (biner 0111)
-	NOT	-10 (desimal) = -12 (desimal)
>>	Digunakan untuk menggeser bit ke kanan dengan	12 (biner 1100) >>4 4 (biner 0100)

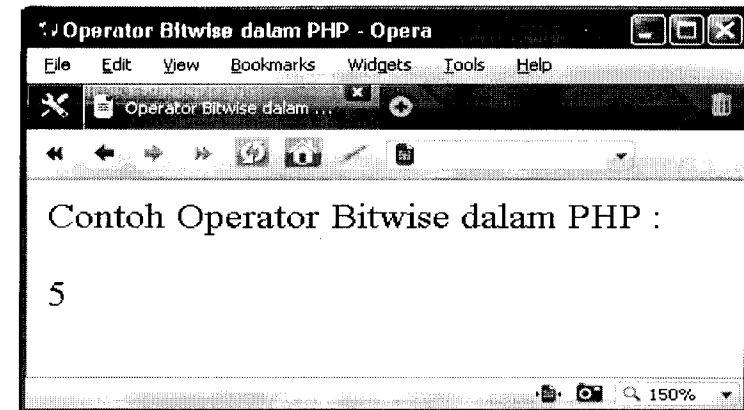
Operator	Arti	Contoh
<<	Digunakan untuk menggeser bit ke kiri dengan	12 (biner 1100) <<4 96 (biner 1100000)

Contoh program:

```

<HTML>
<HEAD>
<TITLE>Operator Bitwise dalam PHP</TITLE>
</HEAD>
<BODY>
Contoh Operator Bitwise dalam PHP : <BR> <BR>
<?php
echo
(
15
&5
);
?>
</BODY>
</HTML>

```



Gambar 4.13 Unary Operator

4.4 EKSPRESI

Ekspresi merupakan bagian yang sangat penting dalam membangun PHP. Formulasi dasar untuk ekspresi adalah konstanta dan variabel. Sebagai contohL: `$nama = "Ani Lisiani"` dikategorikan ke dalam variabel `$nama`, sedangkan "Ani Lisiani" adalah nilai yang berisi "Ani Lisiani", sehingga "Ani Lisiani" memiliki tipe string.

4.4.1 EKSPRESI INCREMENT DAN DECREMENT

Dalam penulisannya, Ekspresi Increment dapat berupa:

`$b = $a++` : definisikan nilai `$b` sama dengan nilai `$a`, dan tingkatkan

(increment) nilai `$a`.

`$b = ++$a` : tingkatkan nilai `$a`, dan definisikan nilai `$b` sama dengan nilai `$a`.

4.4.2 EKSPRESI KONDISIONAL

Ekspresi singkat untuk melakukan kendali program:

`$first ? $second : $third` (jika `$first` benar, maka `$second` menjadi nilai ekspresi, sedangkan jika `$first` salah, maka `$third` menjadi nilai ekspresi).

4.4.3 EKSPRESI PERBANDINGAN

Ekspresi ini menghasilkan nilai 0 atau 1 yang menghasilkan nilai TRUE atau FALSE. PHP mendukung:

Tabel 4.8 Ekspresi Perbandingan

Tanda	Nilai
>	Lebih besar
<	Lebih kecil
>=	Lebih besar sama dengan
<=	Lebih kecil sama dengan
==	Sama dengan
!=	Tidak sama dengan

Seperti pada bahasa C, PHP juga mendukung penulisan ekspresi dalam suatu formulasi, misalnya:

Tabel 4.9 Ekspresi Formulasi

Tanda	Nilai
\$a += 5;	\$a = \$a + 5
\$a -= 10;	\$a = \$a - 10
\$a /= 15	\$a = \$a / 15
\$a *= 20;	\$a = \$a * 20

Pernyataan ekspresi yang lain terlihat dalam penggunaan ternary kondisi yang sifatnya mirip dengan kondisi if ... else, seperti contoh berikut:

```
$merah ? $hijau : $kuning
```

Apabila **\$merah** bernilai TRUE, maka yang akan dikerjakan adalah **\$hijau**; sedangkan jika **\$merah** bernilai FALSE, maka yang akan dikerjakan adalah **\$kuning**.

4.4.4 FUNCTION

Konsep **Function** adalah kumpulan beberapa statement yang dibuat dengan tujuan menyelesaikan satu tugas tertentu.

Return Value dan Parameter

Perhatikan kode berikut:

```
function add($a, $b){
    return $a + $b;
}
```

Function sederhana tersebut akan menerima masukan berupa dua angka, yang akan dijumlahkan, kemudian hasil penjumlahannya akan dikembalikan kepada pemanggil *function*.

Nilai yang dikembalikan tersebut disebut Return Value, sedangkan nilai yang dimasukkan ke dalam *function* (\$a dan \$b) disebut Parameter.

Deklarasi Function

Beberapa hal utama yang perlu diperhatikan dalam deklarasi *function* pada PHP adalah:

- nama function.
- Parameter.
- function body.

Contoh *function*:

```
function addNumber($b, $a)
{
    c = a + b;
    echo(c);
}

function jumlahkanlah(int a, int b)
{
    c = a + b;
```

```
return c;
}
```

Pada contoh *function* **addNumber** tersebut, baris satu merupakan deklarasi *function*, yang berisi:

- keyword *function*.
- nama *function*.
- parameter.

Parameter adalah nilai yang dimasukkan ke dalam *function* untuk diproses sehingga menghasilkan output. Nama *function* ditentukan dengan memenuhi beberapa aturan sebagai berikut, seperti:

- Tidak boleh sama dengan *function* yang terdapat pada PHP.
- Hanya boleh terdiri dari huruf, angka, dan garis bawah (*underscore*).
- Tidak boleh diawali dengan angka.
- `function (){} //` tidak boleh digunakan.

Bahasa pemrograman lain mendukung fasilitas *Overloading*, yaitu sebuah *function* dapat memiliki nama yang sama dan hasil yang berbeda, dengan syarat parameternya berbeda. PHP tidak mendukung fasilitas *Overloading*, sehingga Anda tidak dapat menggunakan nama yang telah digunakan sebelumnya.

4.5 RANGKUMAN

PHP memiliki beberapa aturan sendiri, seperti halnya bahasa pemrograman pada umumnya. Seperti halnya konstanta, yang merupakan variabel bernilai tetap dan suatu identifier untuk nilai yang sederhana. Selain itu, PHP juga menyediakan operator yang digunakan untuk mengetahui operasi apa yang harus dilakukan.

Jenis operator dalam PHP:

- Operator Aritmetika (+, -, *, /, %).
- Operator String (.
- Operator Increment/Decrement (++ , --).
- Operator Bitwise (&, |, ^, ~, >>, <<).
- Operator Pembandingan (==, <, >, <=, >=, !=, <>).

- Operator Logika (and/ &&, or / ||, xor, !).
- Operator Penugasan (+=, -=, /=, %=, |=, ^=, .=).

Konstanta menyatakan nilai yang tetap di dalam program. Penulisannya selalu diapit tanda petik tunggal atau petik ganda; tanda petik tersebut bukan merupakan bagian dari konstanta. Konstanta yang terdiri dari satu karakter (yang dikenal sebagai konstanta karakter) harus diawali dan diakhiri dengan tanda petik tunggal.

4.6 KUIS

1. Apa perbedaan antara konstanta dan variabel?
2. Berapakah jenis operator yang terdapat dalam PHP? Jelaskan!
3. Simbol-simbol apa yang digunakan dalam Operator Perbandingan? Jelaskan beserta fungsinya!
4. Apa perbedaan antara Operator Perbandingan dan Ekspresi Perbandingan?
5. Dalam penggunaan operator aritmatika, dalam mencari luas segitiga (L), apakah "L" bisa diketahui sebelum dideklarasikan? Jelaskan!

4.7 LATIHAN

Buatlah skrip PHP menggunakan operator aritmatika yang menampilkan hasil pada browser, dengan ketentuan sebagai berikut:

Diketahui:

$P = 10$

$L = 5$

Carilah keliling dari sebuah persegi panjang dengan rumus:

$K = P \times L$

Selanjutnya, simpan dengan nama **Persegi_Panjang.php**, setelah itu jalankan pada browser.

Sesi V

Struktur Kontrol PHP

Durasi	2 jam
Jenis	Teori dan praktik.
Media	PHP 5, web browser.
Tujuan	Setelah membaca sesi ini, siswa diharapkan dapat: ❖ Memahami struktur kontrol pada PHP. ❖ Memahami penggunaan struktur kontrol dalam perancangan pemrograman. ❖ Menguasai logika pemrograman dalam PHP.

Berbagai macam bahasa pemrograman (hampir semua) menyediakan fasilitas struktur kontrol, seperti: Basic, Pascal, Turbo C, CGI, C++, C#, dan lainnya. Bahasa pemrograman pada web juga menggunakan struktur kontrol, seperti: ASP, AJAX, PERL, Cold Fusion, PHP, dan lainnya. Pada prinsipnya, penggunaan struktur kontrol hampir sama jika dilihat dari logika pemrograman; perbedaannya terletak pada penulisan sintaks dan tanda yang digunakan untuk membuka dan menutup statemen. Pada dasarnya, PHP dibuat dengan deret dari suatu statemen. Berbagai statement tersebut dapat berupa statementproses perulangan, statement penugasan, statementkondisi, pernyataan dapat berupa assignment dan pemanggilan fungsi. Suatu statement dalam PHP dapat ditandai dengan tanda semikolon(;). Beberapa pernyataan dapat dikelompokkan menjadi suatu kelompok dengan menggunakan tanda kurung kurawal ({ }). Sebuah kelompok pernyataan dapat merupakan sebuah pernyataan juga.

■ IF

Sintaks: if (expr) statement

Contoh: -if (\$x > \$y) print "x lebih besar dari y";

- Jika statement lebih dari satu, maka:

```
if ($x > $y) {
    print "x lebih besar dari y";
    $y = $x;
}
```

■ Else

```
if ($x > $y) {
    print "x lebih besar dari y";
} else {
    print "x tidak lebih besar dari y";
}
```

■ Elseif

```
if ($x > $y) {
    print "x lebih besar dari y";
} elseif ($x == $y) {
```

```
    print "x sama dengan y";
} else {
    print "x lebih kecil dari y";
}
```

■ Switch

```
switch ($n) {
    case 0:
        print "n sama_dengan 0";
    case 1:
        print "n sama_dengan 1";
    case 2:
        print "n sama_dengan 2";
}
```

5.1 PERCABANGAN/BENTUK IF

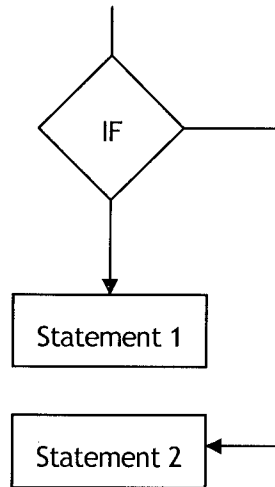
Percabangan - sering disebut dengan istilah **decision-making**- memungkinkan aplikasi untuk memeriksa isi suatu variabel atau hasil perhitungan ekspresi dan mengambil tindakan yang sesuai. Ada dua jenis percabangan yang dipilih berdasarkan kriteria pemeriksaan dan jumlah pilihan yang tersedia. Bentuk ini mempunyai nilai TRUE dan FALSE.

5.1.1 STRUKTUR IF

Sintaks:

```
If(ekspresi) {
    Statement-jika kondisi true;
}
```

Berdasarkan gambaran sintaks tersebut, kita dapat menyimpulkan bahwa statement akan dikerjakan apabila ekspresi bernilai TRUE. Dengan kata lain, statement akan dikerjakan apabila ekspresi bernilai benar, seperti yang digambarkan dalam *flowchart* berikut:



5.1.2 STRUKTUR IF-ELSE

Konstruksi **if-else** dapat dijelaskan sebagai berikut:

```

if(kondisi)
{
    // statement 1 kerjakan
}
else
{
    // statement 2 kerjakan
}
// statement 3 kerjakan
  
```

Keterangan program:

- Langkah 1, kondisi akan diperiksa.
- Langkah 2, bila bernilai TRUE, maka statement 1 akan dikerjakan

- Langkah 3, bila bernilai FALSE, maka statement 2 akan dikerjakan
- Langkah 4, langkah 3 dijalankan

5.1.3 STRUKTUR IF-ELSEIF-ELSE

PHP menyediakan konstruksi **if-elseif-else** untuk membuat pilihan lebih dari dua.

```

if(kondisi1)
{
    // statement 1 kerjakan
}
elseif(kondisi 2) kerjakan
{
    // statement 2 kerjakan
}
else
{
    // statement 3 kerjakan
}
// statement 4 kerjakan
  
```

Keterangan program:

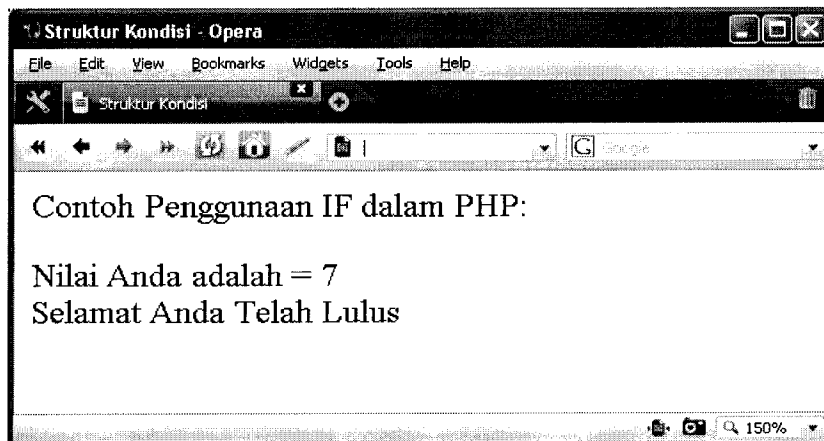
- Jika kondisi 1 bernilai TRUE, maka:
Statement 1 dijalankan
Statement 4 dijalankan
- Apabila kondisi 1 dan kondisi 2 masing-masing bernilai FALSE, maka:
Statement 2 dijalankan
Statement 4 dijalankan
- Apabila kondisi 1 dan kondisi 2 bernilai FALSE, maka:
Statement 3 dijalankan
Statement 4 dijalankan

Contoh Program:

```

<HTML>
<HEAD>
<TITLE>Struktur Kondisi</TITLE>
</HEAD>
<BODY>
Contoh Penggunaan IF dalam PHP: <BR>
<?php
$Nilai=7;
echo ("Nilai Anda adalah = $Nilai <BR>");
if ($Nilai>6)
{
    echo "Selamat Anda Telah Lulus <BR>";
}
?>
</BODY>
</HTML>

```



Gambar 5.1 Struktur IF Tunggal

Program tersebut terdapat pemeriksaan kondisi, apakah isi variabel **\$Nilai** lebih besar dari 6. Jika kondisi ini bernilai TRUE (variabel **\$Nilai** berisi nilai lebih besar dari 6), maka statement pada Selamat Anda Telah Lulus.

Contoh program:

```

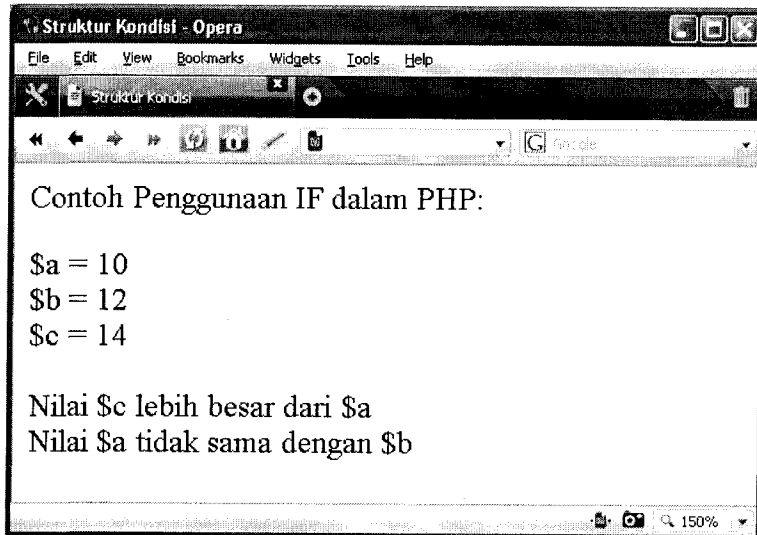
<HTML>
<HEAD>
<TITLE>Struktur Kondisi</TITLE>
</HEAD>
<BODY>
Contoh Penggunaan IF ELSE dalam PHP: <BR>
<?php
$a=10;
$b=15;
$c=18;
Printf ("\a = %d <BR>", $a);
Printf ("\b = %d <BR>", $b);
Printf ("\c = %d <BR><BR>", $c);
if ($c > $a)
{
    echo("Nilai \c lebih besar dari \a<BR>");
}
if ($a == $b)
{
    echo("Nilai \a sama dengan \b");
}
else
{
    echo("Nilai \a tidak sama dengan \b");
}

```

```

}
?>
</BODY>
</HTML>

```



Gambar 5.2 Struktur Kondisi IF ELSE

Pada program tersebut, variabel **\$a**, **\$b**, **\$c** memiliki masing-masing nilai yang berbeda, yang menunjukkan besar nilai tersebut, sehingga dengan menggunakan variabel ketentuan mencari nilai terbesar dapat menggunakan variabel.

Contoh program:

```

<HTML>
<HEAD>
<TITLE> Struktur Kondisi </TITLE>
</HEAD>
<BODY>

```

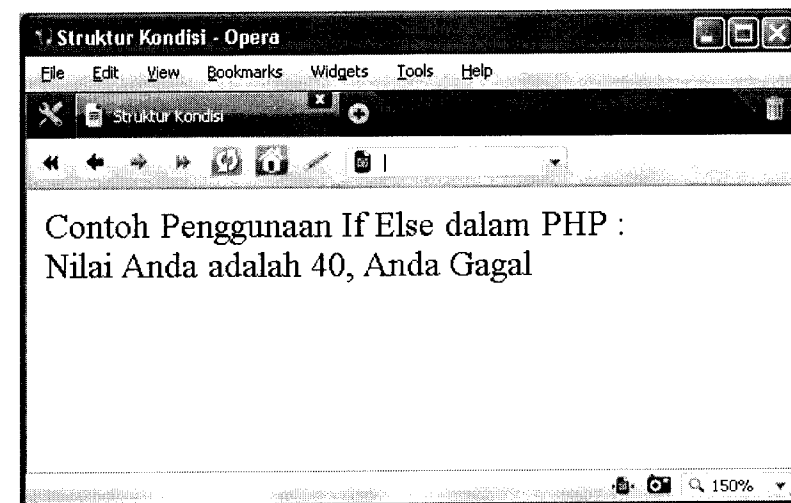
Contoh Penggunaan If Else dalam PHP :


```
<?php
```

```

$Nilai = 40;
if ($Nilai >= 60)
{
    echo "Nilai Anda adalah $Nilai, Anda Lulus";
}
else
{
    echo "Nilai Anda adalah $Nilai, Anda Gagal";
}
?>
</BODY>
</HTML>

```

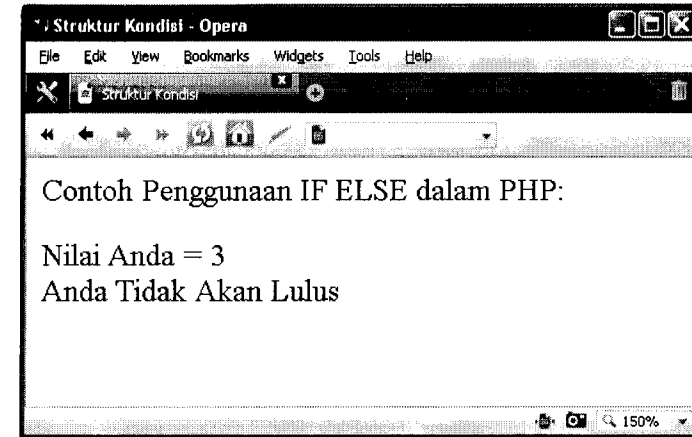


Gambar 5.3 Struktur Kondisi IF ELSE Mencari Nilai

Contoh program tersebut menunjukkan jika menggunakan dua buah kondisi dengan **\$Nilai >= 60**, maka statement pertama "Nilai Anda adalah \$Nilai, Anda Lulus" akan dijalankan. Jika **\$Nilai < 60**, maka statement kedua "Nilai Anda adalah \$Nilai, Anda Gagal" yang akan dijalankan.

Contoh program:

```
<HTML>
<HEAD>
<TITLE>Struktur Kondisi</TITLE>
</HEAD>
<BODY>
Contoh Penggunaan IF ELSE dalam PHP: <BR>
<?php
$Nilai=3;
echo ("Nilai Anda = $Nilai <BR>");
if ($Nilai>6)
{
    echo "Anda Lulus <BR>";
}
else if ($Nilai>=4)
{
    echo "Anda Harus Mengulang <BR>";
}
else
{
    echo "Anda Tidak Akan Lulus <BR>";
}
?>
</BODY>
</HTML>
```



Gambar 5.4 Struktur Kondisi IF ELSE dengan 2 Kondisi

Program tersebut menunjukkan bahwa IF ... ELSE kali ini memiliki dua buah kondisi atau bisa dikatakan IF **jamak**. Pada kondisi tertentu, bukan statement kedua yang dijalankan, melainkan server akan memanggil kondisi berikutnya. Perhatikan penggalan skrip berikut:

```
if ($Nilai>6)
{
    echo "Anda Lulus <BR>";
}
```

Penggalan skrip tersebut menunjukkan bahwa dengan kondisi \$Nilai > 6, maka akan muncul statement "Anda Lulus". Namun demikian, akan berbeda apabila Anda memasukkan kondisi lain jika kondisi pertama tidak terpenuhi "else if (\$Nilai>=4)", dapat dilihat dengan kondisi \$Nilai >= 4, maka "Anda harus mengulang".

```
else if ($Nilai>=4)
{
    echo "Anda Harus Mengulang <BR>";
}
```

Pada penggalan skrip tersebut merupakan penggunaan kondisi kedua sebagai alternatif jika kondisi pertama tidak terpenuhi. Dengan memberikan range antara 4 <= \$Nilai >=6, maka statement kedua akan

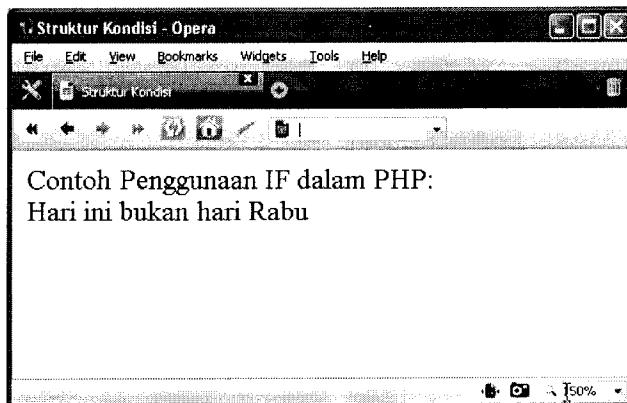
dipanggil “Anda Harus Mengulang”. Setelah kedua kondisi tersebut tidak terpenuhi, maka statement ketiga yang akan dipanggil “Anda Tidak Akan Lulus”.

Contoh program:

```
<HTML>
<HEAD>
<TITLE>Struktur Kondisi</TITLE>
</HEAD>
<BODY>
```

Contoh Penggunaan IF dalam PHP:


```
<?php
    $hari = date("w");
    if ($hari == 0)
        echo("Hari ini hari Rabu");
    else
        echo("Hari ini bukan hari Rabu");
?>
</BODY>
</HTML>
```



Gambar 5.5 Struktur Kondisi IF ELSE Menggunakan Fungsi DATE

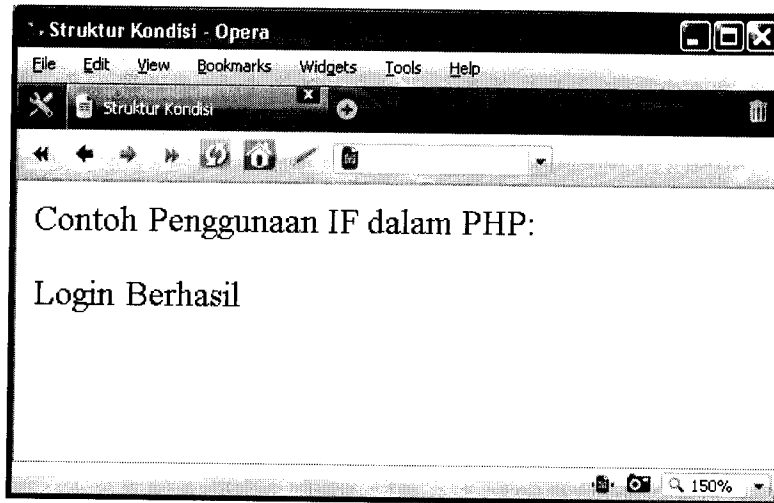
Selain untuk mencari perbandingan dari suatu nilai IF ... ELSE, juga dapat digunakan untuk mencari keterangan hari. Dengan menggunakan fungsi **date()** yang tersedia pada PHP. Contoh skrip tersebut menerangkan kondisi 0, berarti FALSE. Dengan demikian, maka server akan memanggil statement kedua “Hari ini bulan hari Rabu”.

Contoh program:

```
<HTML>
<HEAD>
<TITLE>Struktur Kondisi</TITLE>
</HEAD>
<BODY>
```

Contoh Penggunaan IF dalam PHP:


```
<P>
<?php
    $user="KrabyPatty";
    $pass="adadeh";
    if ($user == " KrabyPatty" && $pass == "adadeh") {
        echo "Login Berhasil";
    } else {
        echo "Login Gagal";
    }
?>
</BODY>
</HTML>
```



Gambar 5.6 Memeriksa Username dan Password dengan if...else

Pada program tersebut menggunakan pemeriksaan kondisi - apakah isi variabel `$user` sama dengan "KrabyPatty" dan apakah isi variabel `$pass` sama dengan "adadeh". Jika kedua kondisi bernilai TRUE, maka tampilan pada browser akan menyatakan Login Berhasil. Jika kondisi bernilai FALSE, maka browser akan menampilkan Login Gagal.

5.1.4 STRUKTUR SWITCH...CASE

Switch Case digunakan sebagai alternatif dari struktur if...else; dengan else lebih dari satu. Dengan statement ini, program akan semakin mudah dibuat karena peran switch digunakan pada saat ditemukannya suatu masalah dan membandingkan suatu variabel dengan berbagai nilai. Struktur Switch Case sangat cocok digunakan untuk kondisi logika yang memiliki nilai-nilai pasti dalam logika tersebut.

Perhatikan struktur logika Switch Case berikut ini:

```
switch(n){
case value_case1;
// statement 1 dijalankan
break;
case value_case2;
```

```
// statement 2 dijalankan
break;
case value_case3;
// statement 3 dijalankan
break;
default;
// statement 4 dijalankan
break;
}
// statement 5 dijalankan
```

Aliran program:

1. Variabel `n` diperiksa.
2. Statement dieksekusi.
 - `n == 1`, statement 1 dijalankan.
 - `n == 2`, statement 2 dijalankan.
 - `n == 3`, statement 3 dijalankan.
 - `n` tidak memenuhi `2n-2o`, statement 4 dijalankan.
3. Statement 5 dijalankan.

Contoh program:

Contoh berikut ini merupakan pembuatan pencarian tanggal menggunakan Switch Case:

```
<HTML>
<HEAD>
<TITLE>Struktur Kondisi</TITLE>
</HEAD>
<BODY>
Contoh Penggunaan SWITCH CASE dalam PHP : <BR>
<P>
<?php
```

```

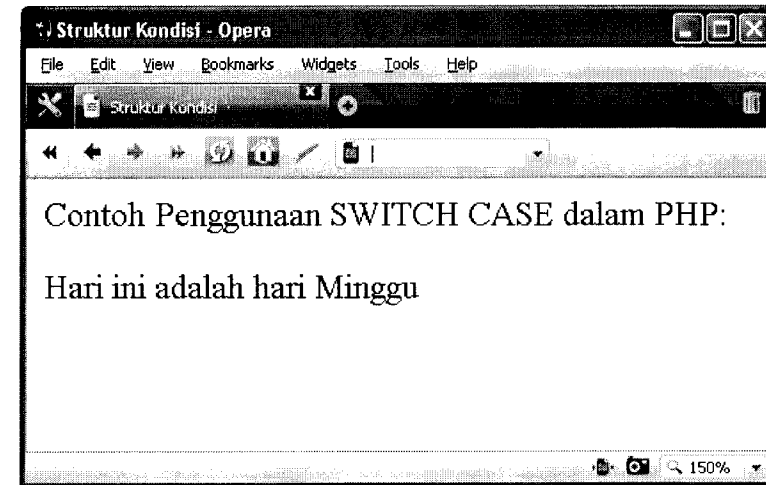
$hari = date("1");
switch($hari)
{
case "Monday":
    $harinya = "Senin";
    break;
case "Tuesday":
    $harinya = "Selasa";
    break;
case "Wednesday":
    $harinya = "Rabu";
    break;
case "Thursday":
    $harinya = "Kamis";
    break;
case "Friday":
    $harinya = "Jumat";
    break;
case "Saturday":
    $harinya = "Sabtu";
    break;
default :
    $harinya = "Minggu";
    break;
}
printf("Hari ini adalah hari $harinya");
?>

```

```

</BODY>
</HTML>

```



Gambar 5.7 Struktur Kondisi SWITCH CASE Mencari Keterangan Hari dengan Fungsi date()

Pada program tersebut, variabel `$hari` menggunakan fungsi `date()`, sehingga Anda akan memperoleh informasi berupa tanggal, hari, dan waktu sekarang, kemudian variabel `$hari` akan diperiksa dengan Switch jika dimasukkan angka "1", maka `$hari`-nya adalah "Minggu".

Contoh program:

```

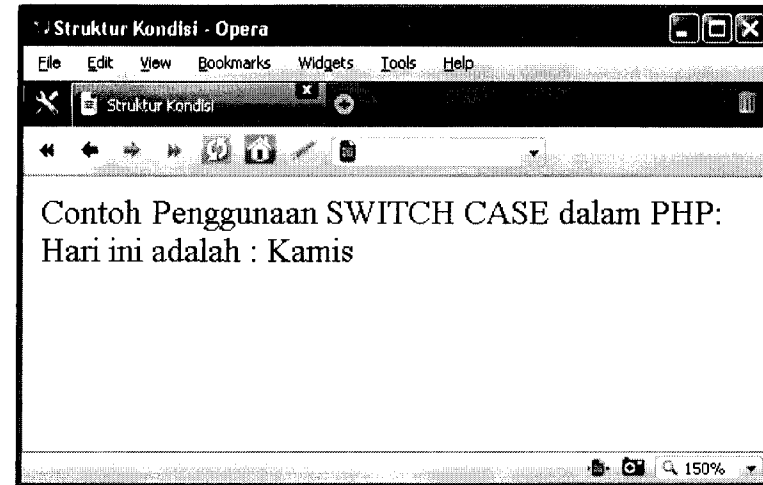
<HTML>
<HEAD>
<TITLE>Struktur Kondisi</TITLE>
</HEAD>
<BODY>
Contoh Penggunaan SWITCH CASE dalam PHP: <BR>
Hari ini adalah :
<?php
    $hari = date("w");

```

```

switch($hari)
{
    case 0:
        echo("Minggu");
        break;
    case 1:
        echo("Senin");
        break;
    case 2:
        echo("Selasa");
        break;
    case 3:
        echo("Rabu");
        break;
    case 4:
        echo("Kamis");
        break;
    case 5:
        echo("Jumat");
        break;
    default:
        echo("Sabtu");
}
?>
</BODY>
</HTML>

```



Gambar 5.8 Struktur Kondisi SWITCH CASE 7 Hari dengan Fungsi date()

Berbeda dengan contoh program sebelumnya, program tersebut menggunakan SWITCH CASE untuk semua nama hari, sedangkan yang menjadi **default()** adalah hari Sabtu, sehingga apabila CASE 1 hingga CASE 5 tidak terpenuhi, maka server akan memanggil **default()**, yaitu hari Sabtu.

Contoh lainnya, skrip untuk menampilkan nama hari dalam bahasa Indonesia menggunakan *switch case* yang lebih sederhana.

```

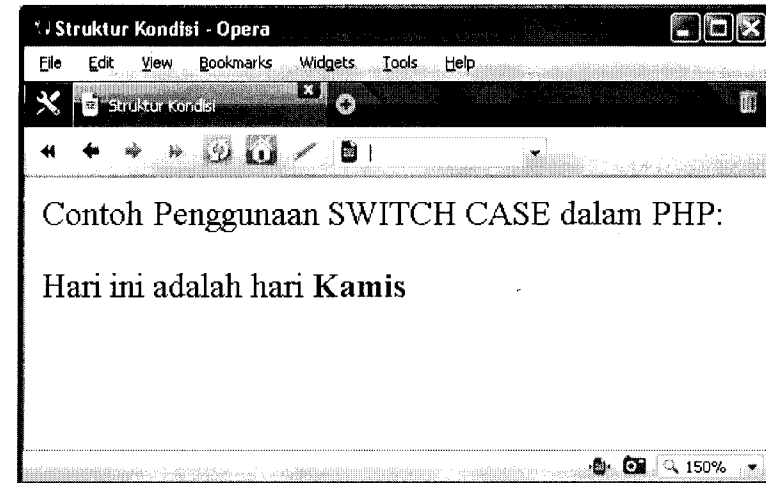
<HTML>
<HEAD>
<TITLE>Struktur Kondisi</TITLE>
</HEAD>
<BODY>
Contoh Penggunaan SWITCH CASE dalam PHP: <BR>
<P>
<?php
$day = date("D");
switch ($day) {
    case 'Sun'           : $hari = "Minggu";

```

```

break;
case 'Mon'           : $hari = "Senin";
break;
case 'Tue'           : $hari = "Selasa";
break;
case 'Wed'           : $hari = "Rabu";
break;
case 'Thu'           : $hari = "Kamis";
break;
case 'Fri'           : $hari = "Jum'at";
break;
case 'Sat'           : $hari = "Sabtu";
break;
default              : $hari = "Hari Akhir";
}
printf("Hari ini adalah hari <b>$hari</b>");
?>
</BODY>
</HTML>

```



Gambar 5.9 Struktur Kondisi SWITCH CASE 7 Hari dengan Konversi Hari

Contoh program:

Program berikut digunakan untuk mencari keterangan nilai yang diwakili dengan huruf dengan menggunakan struktur *switch case* dengan kondisi sebagai berikut:

- Jika nilai huruf = "A", maka keterangan nilai = "Memuaskan"
- Jika nilai huruf = "B", maka keterangan nilai = "Baik"
- Jika nilai huruf = "C", maka keterangan nilai = "Cukup"
- Jika nilai huruf = "D", maka keterangan nilai = "Kurang"
- Jika nilai huruf = "E", maka keterangan nilai = "Gagal"

```

<HTML>
<HEAD>
<TITLE>Struktur Kondisi</TITLE>
</HEAD>
<BODY>

```

Contoh Penggunaan SWITCH CASE dalam PHP :


```

<?php
$nilaihuruf = ("C");

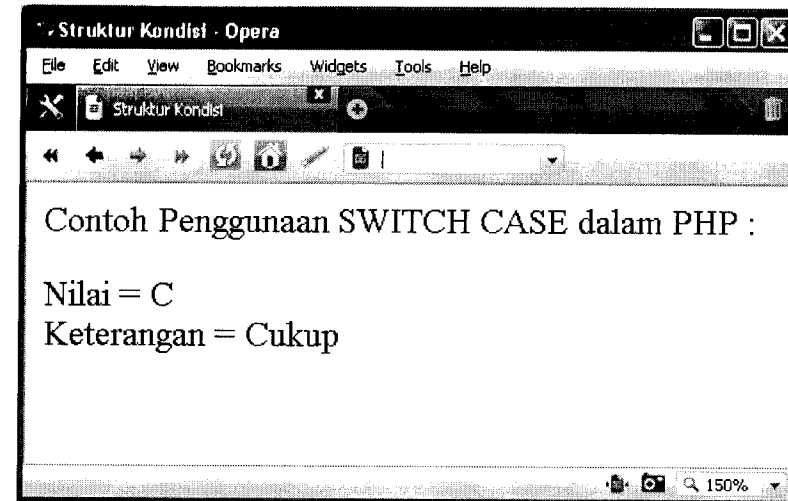
```

```

switch($nilaihuruf)
{
case "A":
    $keterangan = "Memuaskan";
    break;
case "B":
    $keterangan = "Baik";
    break;
case "C":
    $keterangan = "Cukup";
    break;
case "D":
    $keterangan = "Kurang";
    break;
default :
    $keterangan = "Gagal";
    break;
}
echo "Nilai = $nilaihuruf<br>";
echo "Keterangan = $keterangan";

?>
</BODY>
</HTML>

```



Gambar 5.10 Struktur Kondisi SWITCH CASE Mencari Keterangan \$Nilai

Penggunaan *switch case* pada program tersebut digunakan untuk mencari keterangan dari suatu nilai dari variabel *\$nilaihuruf*. Nilai dalam *\$nilaihuruf* adalah "C". Perhatikan penggalan skrip berikut ini:

```

$nilaihuruf = ("C");
switch($nilaihuruf)
{
case "A":
    $keterangan = "Memuaskan";
    break;
case "B":
    $keterangan = "Baik";
    break;
case "C":
    $keterangan = "Cukup";
    break;
case "D":

```

```

    $keterangan = "Kurang";
    break;
default :
    $keterangan = "Gagal";
    break;
}

```

Server akan membaca skrip `$nilaihuruf="C"`, jika salah satu *case* `$nilaihuruf="C"`, maka bernilai **TRUE** dan statement dijalankan. Jika bernilai **FALSE**, maka sistem akan berhenti (`break;`) kembali ke kondisi pertama, kemudian ke statement berikutnya atau kedua akan dijalankan. **SWITCH CASE** akan berjalan terus sampai kondisi terpenuhi **TRUE**.

5.2 LOOP/PERULANGAN

Loop merupakan proses eksekusi operasi program secara berulang-ulang sampai ditemui kondisi/batasan untuk mengakhiri eksekusi tersebut. Pada umumnya, loop berisi kode/kumpulan kode yang akan diproses berulang-ulang. Dalam hal ini, Anda tidak perlu menulis ulang sebuah perintah sebanyak pengulangan yang diinginkan. Karena proses terjadi berulang-ulang, maka ada beberapa hal yang perlu diperhatikan sebagai dasar dari suatu proses *loop*. **Pertama**, menentukan kapan dimulainya suatu perulangan dan kapan suatu proses perulangan itu harus diakhiri. **Kedua**, berapa step yang dibutuhkan untuk suatu proses perulangan; step dalam proses perulangan dapat bernilai positif atau negatif.

PHP memiliki 3 jenis perintah *loop* (perulangan), namun dalam perkembangannya menjadi lebih banyak, seperti pada penjelasan berikut.

5.2.1 For

Perulangan **FOR** berfungsi untuk mengulangi perintah dengan jumlah perulangan yang sudah diketahui. Pada perintah ini, Anda tidak perlu menuliskan sebuah kondisi untuk diuji. Anda hanya menuliskan nilai awal dan akhir variabel perhitungan. Nilai ini akan bertambah dan berkurang secara otomatis setiap kali melaksanakan sebuah pengulangan.

Sintaknya adalah:

```

for (nilai_awal, nilai_akhir, peningkatan/penurunan)
{
    pernyataan yang akan dijalankan
}

```

Perhatikan contoh berikut ini:

```

<HTML>
<HEAD>
<TITLE>Struktur Perulangan</TITLE>
</HEAD>
<BODY>

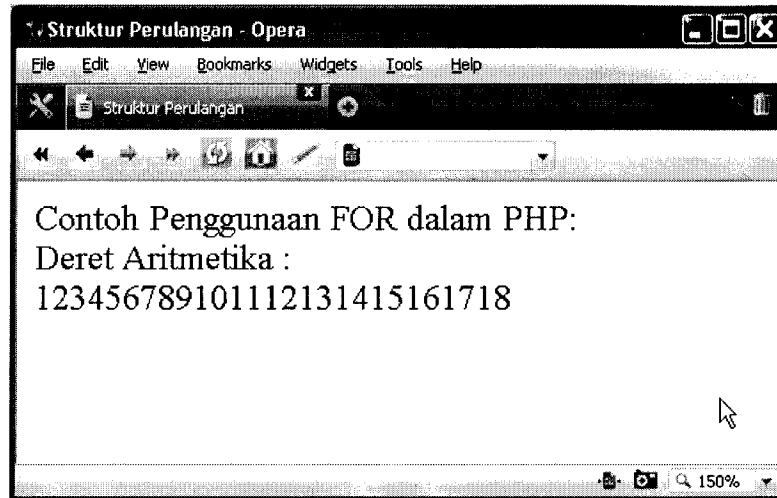
```

Contoh Penggunaan FOR dalam PHP: `<BR>`

```

<?php
echo "Deret Aritmetika : <br>";
for ($n = 1; $n <= 18; $n++)
{
    print $n;
}
?>
</BODY>
</HTML>

```



Gambar 5.11 Struktur FOR pada Deret Aritmatika 1-18

Berbeda dengan kondisi sebelumnya, ketika variabel dideklarasikan dengan suatu nilai. Untuk perulangan/looping, justru variabel merupakan proses dari suatu statement yang nantinya akan ditampilkan, seperti penggalan skrip berikut:

```
for ($n = 1; $n <= 18; $n++)
```

Variabel `$n` dideklarasikan terlebih dahulu dengan *range* tertentu dengan `$n++`. Nilai `n` yang akan ditampilkan antara 1 sampai 18. Setelah menentukan perulangan yang akan dijalankan oleh variabel `$n`, statement `$n` digunakan untuk menampilkan proses *looping* tersebut.

```
{
    print $n;
}
```

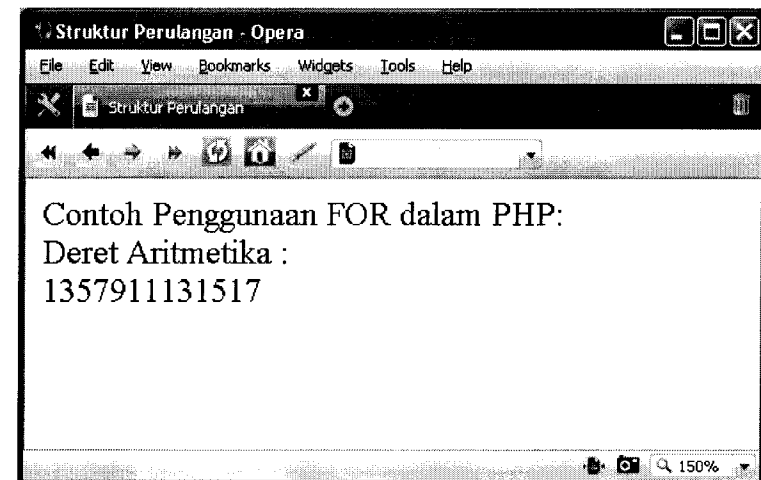
Setelah `$n` dideklarasikan dan diberi range untuk proses looping dengan perintah `print $n`, maka server akan memrosesnya untuk menampilkannya pada browser.

Contoh program:

Menampilkan bilangan ganjil dengan menggunakan FOR.

```
<HTML>
```

```
<HEAD>
<TITLE>Struktur Perulangan</TITLE>
</HEAD>
<BODY>
Contoh Penggunaan FOR dalam PHP: <BR>
<?php
echo "Deret Aritmetika Menampilkan Bilangan Ganjil: <br>";
for ($n = 1; $n <= 18; $n+=2)
{
    print $n;
}
?>
</BODY>
</HTML>
```



Gambar 5.12 Struktur FOR Deret Aritmatika Bilangan +2

Proses perulangan/looping pada Gambar 5.12 sama dengan proses perulangan sebelumnya. Kali ini, proses yang digunakan tidak dengan `$n++`, melainkan `$n+2` yang tujuannya setiap satu bilangan ditambah

dengan 2 angka. Jika dimulai dari angka satu, maka angka yang muncul berikutnya adalah: 3, 5, 7, dan seterusnya. Dengan demikian, maka bilangan yang akan muncul adalah bilangan ganjil, tetapi jika dimulai dari angka 2 atau genap, maka yang akan muncul adalah bilangan genap.

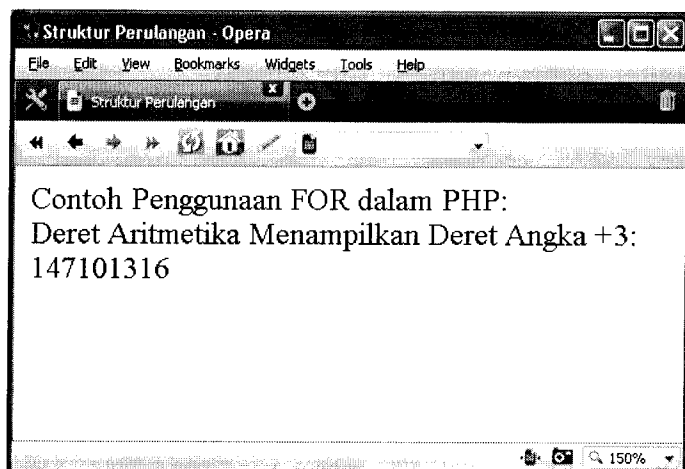
```
for ($n = 1; $n <= 18; $n+=2)
```

Contoh program:

```
<HTML>
<HEAD>
<TITLE>Struktur Perulangan</TITLE>
</HEAD>
<BODY>
```

Contoh Penggunaan FOR dalam PHP:


```
<?php
echo "Deret Aritmetika Menampilkan Deret Angka +3: <br>";
for($n=1;$n<=18;print"$n",$n+=3)
?>
</BODY>
</HTML>
```



Gambar 5.13 Struktur FOR Deret Aritmatika Bilangan +3

Contoh program pada **Gambar 5.13** menggunakan *looping* untuk mencari bilangan genap melalui variabel `$n`. Sama dengan contoh program sebelumnya yang mencari bilangan ganjil, proses perulangan pada program tersebut menggunakan *range* pada nilai variabel yang telah dideklarasikan terlebih dahulu.

Perhatikan penggalan sintaks berikut ini:

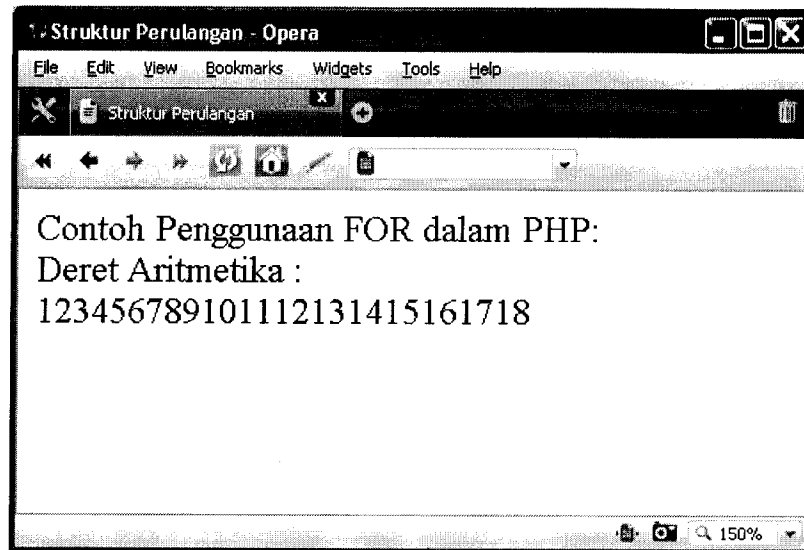
```
for($n=1;$n<=18;print"$n",$n+=3)
```

Contoh program:

```
<HTML>
<HEAD>
<TITLE>Struktur Perulangan</TITLE>
</HEAD>
<BODY>
```

Contoh Penggunaan FOR dalam PHP:


```
<?php
echo "Deret Aritmetika : <br>";
for ($n = 1;;$n++)
{
    if($n>18)
    {
        break;
    }
    print "$n";
}
?>
</BODY>
</HTML>
```



Gambar 5.14 Struktur FOR Deret Aritmatika dengan if

Hampir sama dengan contoh bentuk perulangan pada Gambar 5.11, namun dalam hal ini Anda menggunakan struktur kondisi IF. Seperti biasa, variabel \$n dideklarasikan terlebih dahulu.

```
for ($n = 1;;$n++)
```

Contoh sintaks tersebut tidak menggunakan batasan perulangan yang akan dimunculkan pada browser. Namun demikian, jika menggunakan struktur kondisi IF, maka batasan dalam proses perulangan akan dideklarasikan dalam kondisi IF tersebut.

```
{
    if($n>18)
    {
```

Pada variabel \$n>18, maka statement yang ditampilkan pada browser adalah bilangan antara 1 sampai 18.

Contoh program:

```
<HTML>
<HEAD>
```

```
<TITLE>Struktur Perulangan</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

Contoh Penggunaan FOR dalam PHP:


```
<?php
```

```
echo "Deret Aritmetika : <br>";
```

```
$n=1;
```

```
    for(;;)
```

```
    {
```

```
        if ($n>18)
```

```
        {
```

```
            break;
```

```
        }
```

```
        print "$n";
```

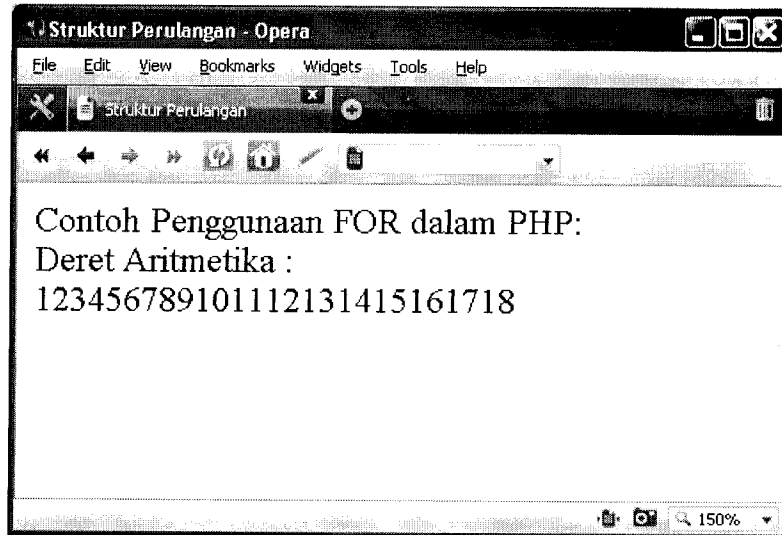
```
        $n++;
```

```
    }
```

```
?>
```

```
</BODY>
```

```
</HTML>
```



Gambar 5.15 Struktur FOR Deret Aritmatika dengan if

Contoh perulangan menampilkan bilangan aritmatika berurutan yang dimulai dengan angka 1. Sebagai penanda yaitu dengan menggunakan variabel `$n=1`. Pada perulangannya menggunakan rumus `$n++`, sedangkan range yaitu antara bilangan 1 sampai 18 if (`$n>18`).

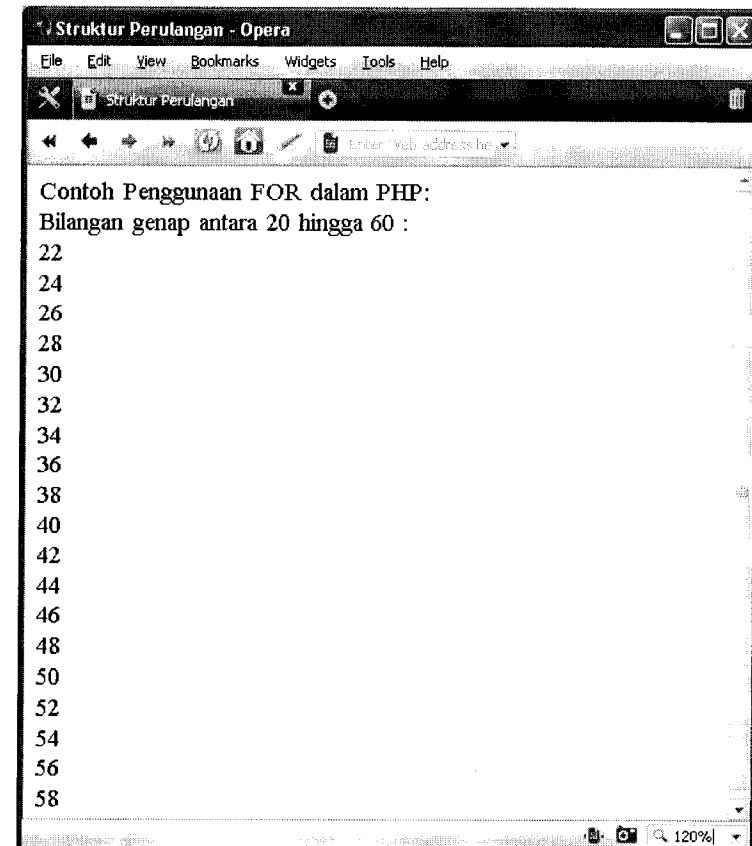
Contoh program:

```
<HTML>
<HEAD>
<TITLE>Struktur Perulangan</TITLE>
</HEAD>
<BODY>
```

Contoh Penggunaan FOR dalam PHP:


```
<?php
$bil = 20;
$cek = 1;
echo ("Bilangan genap antara 20 hingga 60 : <BR>");
for ($cek=1;$cek<20;$cek++)
```

```
{
$bil +=2; // nilai $bil dinaikkan (ditambah) 2
echo (" $bil <BR>");
}
?>
</BODY>
</HTML>
```



Gambar 5.16 Struktur FOR Deret Aritmatika \$bil + 2

Perulangan tersebut menggunakan dua buah variabel, yaitu:

```
$bil = 20;
```

```
$cek = 1;
```

Variabel **\$bil** digunakan untuk menampilkan bilangan yang akan ditampilkan pada browser, sedangkan variabel **\$cek** digunakan untuk membatasi bilangan yang akan diproses oleh server.

```
for ($cek=1;$cek<20;$cek++)
```

\$cek++ merupakan penanda bahwa bilangan akan muncul berurutan, sedangkan **\$bil +=2;**, mengindikasikan bahwa bilangan yang akan ditampilkan merupakan penambahan dua.

Contoh program:

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Struktur Perulangan</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
Contoh Penggunaan FOR dalam PHP: <BR>
```

```
<?php
```

```
function php($teks, $jumlah = 1) {
```

```
    for ($i = 1; $i <= $jumlah; $i++)
```

```
        echo($teks);
```

```
}
```

```
php("..PHP..", 7);
```

```
echo("<BR>\n");
```

```
php("+++"); // sama dengan php("+++",1);
```

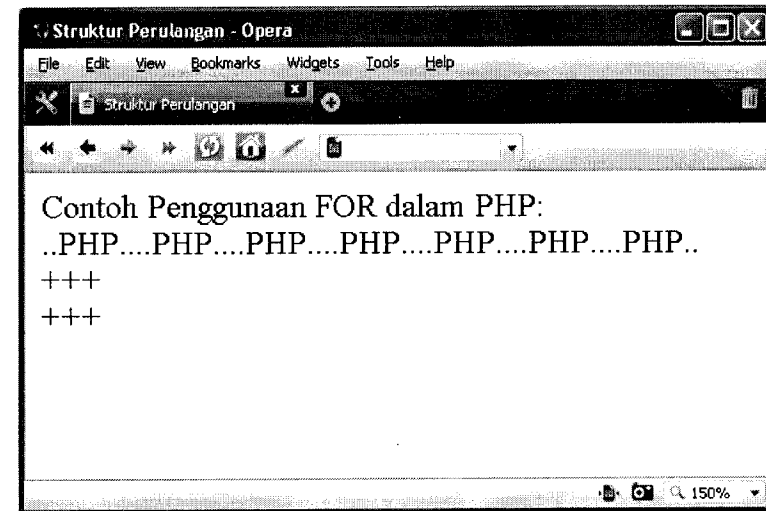
```
echo("<BR>\n");
```

```
php("+++", 1);
```

```
?>
```

```
</BODY>
```

```
</HTML>
```



Gambar 5.17 Struktur FOR Menggunakan FUNCTION

Contoh program:

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Struktur Perulangan</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
Contoh Penggunaan FOR dalam PHP: <BR>
```

```
<?php
```

```
for($n=1;$n<=16;$n+=2):
```

```
    print "Belajar PHP sangat mudah<br>";
```

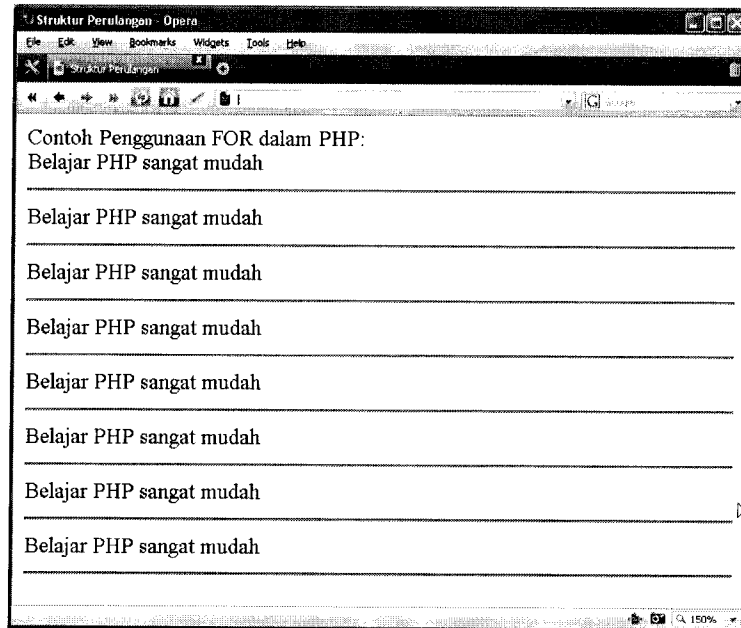
```
    print "<hr color=red size=2>";
```

```
endfor;
```

```
?>
```

```
</BODY>
```

```
</HTML>
```



Gambar 5.18 Struktur FOR Menggunakan Tag HTML <HR>

Contoh program:

```
<HTML>
```

```
<HEAD>
```

```
<TITLE> Perulangan FOR </TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
Contoh Penggunaan FOR dalam PHP : <BR>
```

```
Pilih angka dari 1 sampai 13:
```

```
<P>
```

```
<?php
```

```
echo "<select name='pilih'>\n";
```

```
for ($n=1; $n<=13; $n++)
```

```
{
```

```
    echo ("\t<option value=$n>$n</option>\n");
```

```
}
```

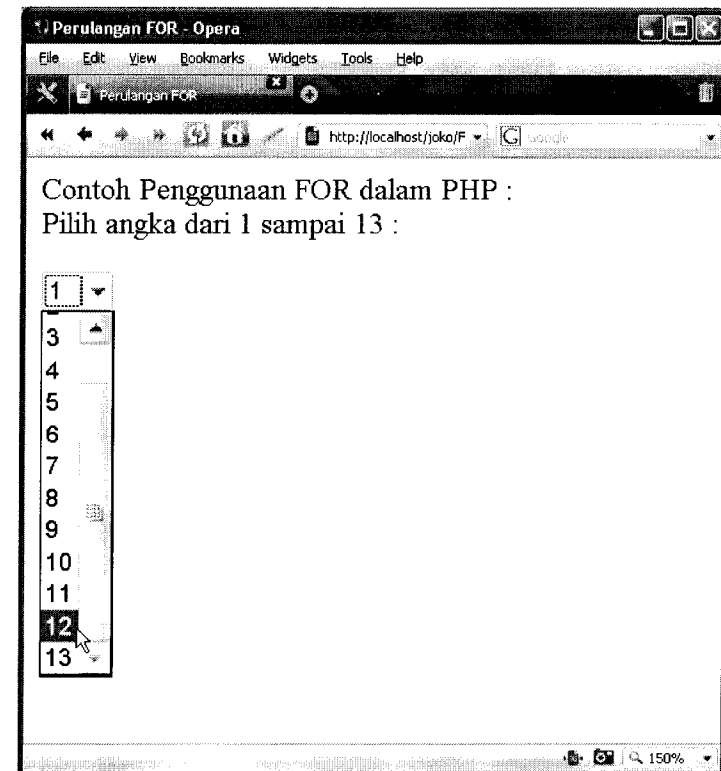
```
echo "</select>\n";
```

```
?>
```

```
</P>
```

```
</BODY>
```

```
</HTML>
```

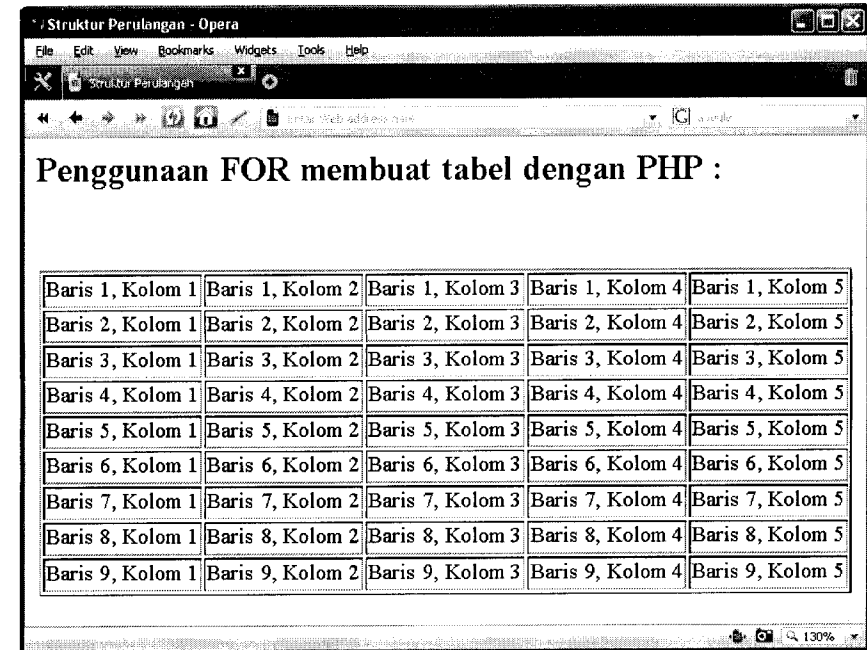


Gambar 5.19 Struktur FOR dengan <OPTION VALUE>

Contoh program:

```
<HTML>
<HEAD>
<TITLE>Struktur Perulangan</TITLE>
</HEAD>
<BODY>
<H2>Penggunaan FOR membuat tabel dengan PHP : </H2>
<BR> <BR>
<center><table border=1>
<?php
for($baris=1;$baris<=9;$baris++)
{
?>
<tr>
<?php
for($kolom=1;$kolom<=5;$kolom++)
{
?>
<td>
<?php
echo "Baris $baris, Kolom $kolom";
?>
</td>
<?php
}
?>
</tr>
<?php
```

```
}
?>
</table></center>
</BODY>
</HTML>
```



Gambar 5.20 Struktur FOR untuk Membuat Tabel

5.2.2 FOREACH

Foreach adalah perintah yang digunakan untuk melakukan iterasi melalui suatu array dengan mudah.

Contoh program:

```
<HTML>
<HEAD>
<TITLE>Struktur Perulangan</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

Contoh Penggunaan FOREACH dalam PHP:


```
<?php
```

```
$example[0] = "Andi Offset";
```

```
$example[1] = "Wahana Komputer";
```

```
$example["Negara"] = "Indonesia";
```

```
$example["KOTA"] = "Kendal";
```

```
$example["no"] = 6172;
```

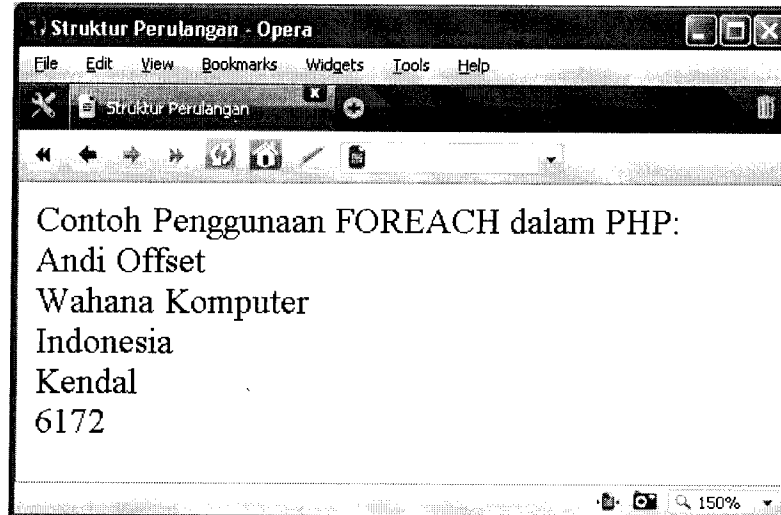
```
foreach($example as $kategori)
```

```
    print($kategori . "<BR>\n");
```

```
?>
```

```
</BODY>
```

```
</HTML>
```



Gambar 5.21 Struktur Looping FOREACH

Contoh program:

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Struktur Perulangan FOREACH</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

Contoh Penggunaan FOREACH dalam PHP:


```
<?php
```

```
$huruf=array(a,b,c,d,e);
```

```
foreach($huruf as $nilai)
```

```
{
```

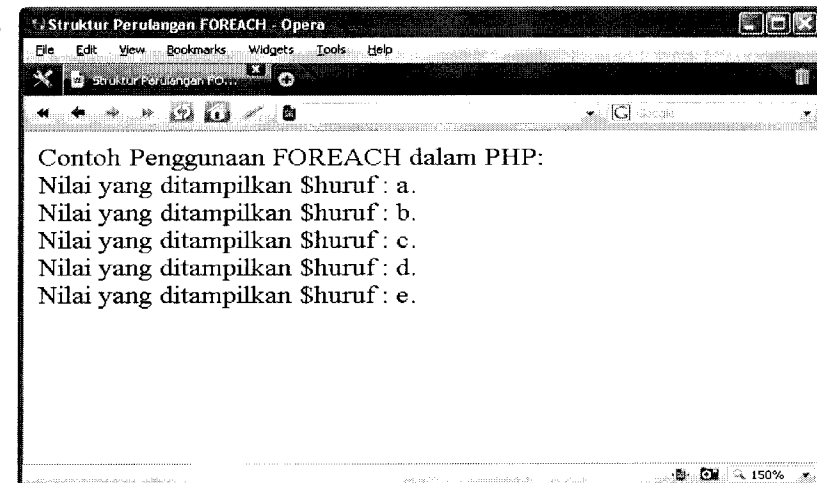
```
    print "Nilai yang ditampilkan \ $huruf : $nilai.<br>\n";
```

```
}
```

```
?>
```

```
</BODY>
```

```
</HTML>
```



Gambar 5.22 Struktur Looping FOREACH dengan ARRAY

5.2.3 WHILE

Perulangan While merupakan perulangan yang paling sederhana dalam PHP. Bentuk dasar pernyataan While adalah:

```

Nilai_awal;
While (kondisi) {
    Statement; //yang akan diulang
    Jumlah;
}

```

Keterangan:

- **Nilai_awal** : nilai awal dari variabel yang digunakan.
- **Statement** : kondisi yang akan membatasi perulangan.
- **Jumlah** : penambahan atau pengurangan nilai variabel, sehingga nilai variabel yang digunakan akan terus berjalan.

Pada While, PHP mengeksekusi pernyataan-pernyataan bersarang (nested statement(s)) berulang-ulang, dengan ketentuan bahwa ekspresi yang dievaluasi bernilai benar (TRUE). Nilai ekspresi tersebut diperiksa setiap saat di awal perulangan. Jika hasil evaluasi ekspresi adalah salah (FALSE) sejak awal, maka pernyataan-pernyataan bersarang tersebut tidak akan dijalankan -meskipun hanya sekali.

Contoh program:

```

<HTML>
<HEAD>
<TITLE> Perulangan WHILE </TITLE>
</HEAD>
<BODY>

```

Contoh Penggunaan WHILE dalam PHP :


```

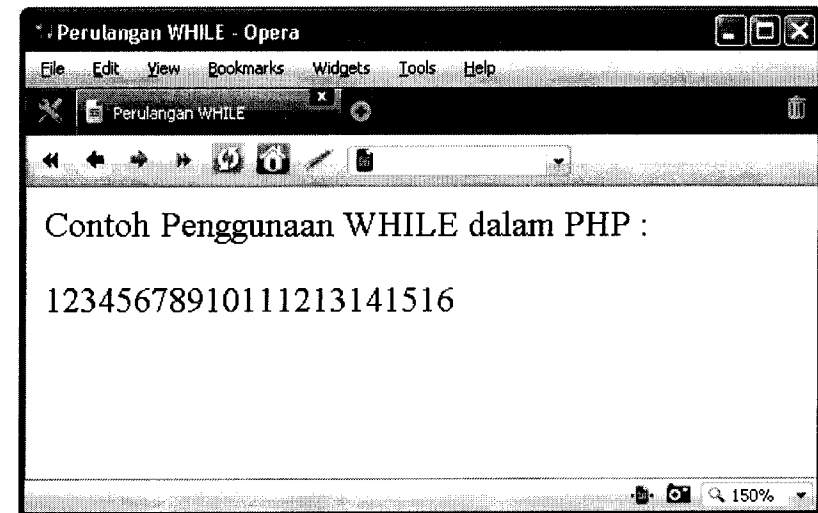
<P>
<?php
$i = 1;
while ($i <= 16)

```

```

{
    print $i++; /* nilai yang akan dicetak $i
    sebelum nilainya naik
    (post-increment) */
}
?>
</P>
</BODY>
</HTML>

```



Gambar 5.23 Struktur Looping FOREACH dengan WHILE

Contoh program:

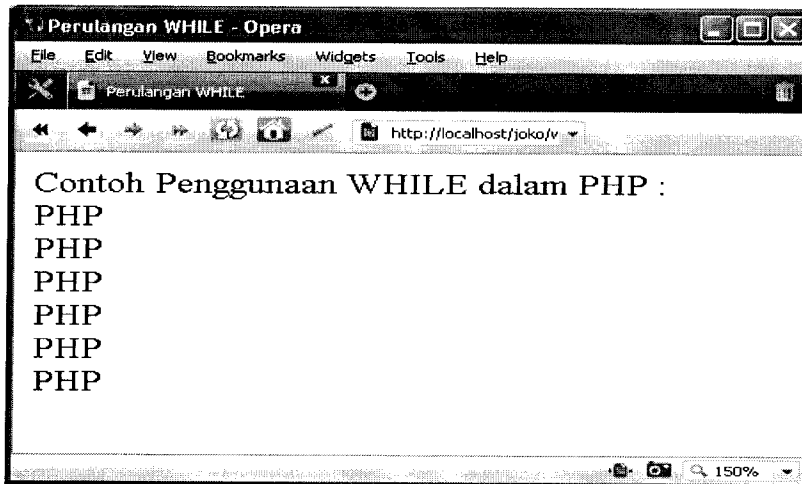
```

<HTML>
<HEAD>
<TITLE>Perulangan WHILE</TITLE>
</HEAD>
<BODY>

```

Contoh Penggunaan WHILE dalam PHP :


```
<?php
    $php = 0;
    while ($php < 6) {
        print("PHP<BR>\n");
        $php++;
    }
?>
</BODY>
</HTML>
```



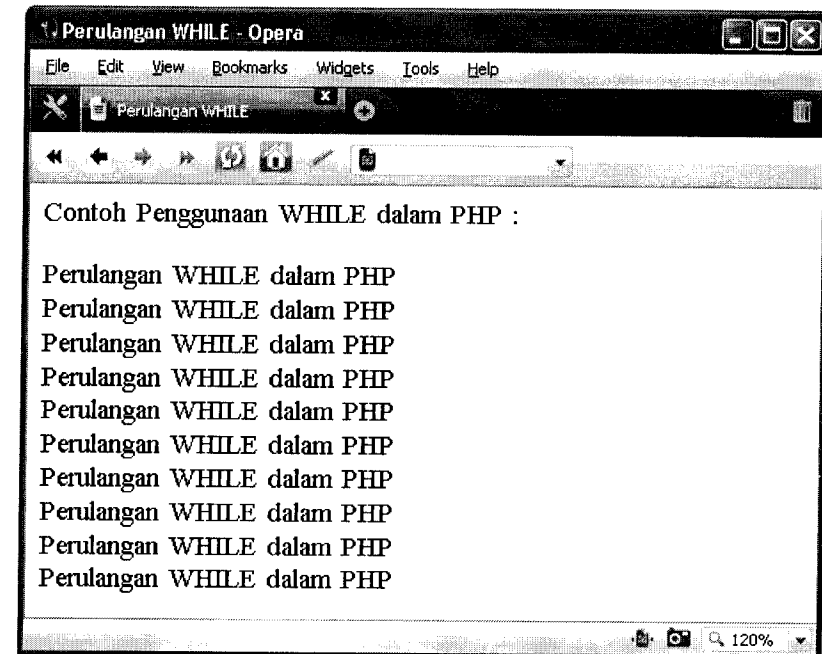
Gambar 5.24 Struktur Looping dengan WHILE

Contoh program:

```
<HTML>
<HEAD>
<TITLE>Menyisipkan PHP di Dokumen HTML</TITLE>
</HEAD>
<BODY>
```

Contoh Penggunaan WHILE dalam PHP :


```
<P>
<?php
    $n = 1;
    while ($n <= 10)
    {
        print"Perulangan WHILE dalam PHP<br>\n";
        $n++;
    }
?>
</P>
</BODY>
</HTML>
```



Gambar 5.25 Struktur FOR dengan WHILE

5.2.4 DO...WHILE

Pada dasarnya, bentuk perulangan **do...while** dengan **While** hampir sama. Perbedaannya terletak pada eksekusi operasi program. Pada pernyataan **While**, maka persyaratan terlebih dahulu diperiksa. Jika persyaratan bernilai **True**, maka proses eksekusi dapat dijalankan, tetapi pada proses **do...while**, operasi program dieksekusi terlebih dahulu kemudian persyaratannya diperiksa. Jika persyaratannya **True**, maka loop/perulangan akan terus dijalankan, perulangan hanya akan berhenti jika proses memiliki hasil **False**. Oleh karenanya, operasi program minimal dapat dieksekusi sekali.

Sintaks dasarnya adalah:

```

Nilai_awal;

Do
{
statement;
jumlah;
}
While(kondisi);

```

Keterangan:

- **Nilai_awal** : merupakan nilai awal variabel.
- **Statement** : statement kondisi yang membatasi perulangan/looping.
- **Jumlah** : merupakan penambahan atau pengurangan nilai suatu variabel agar perulangan tetap berjalan.

Untuk lebih jelasnya, perhatikan contoh pada **Gambar 5.26**.

```

<HTML>
<HEAD>
<TITLE> Perulangan DO WHILE </TITLE>
</HEAD>
<BODY>

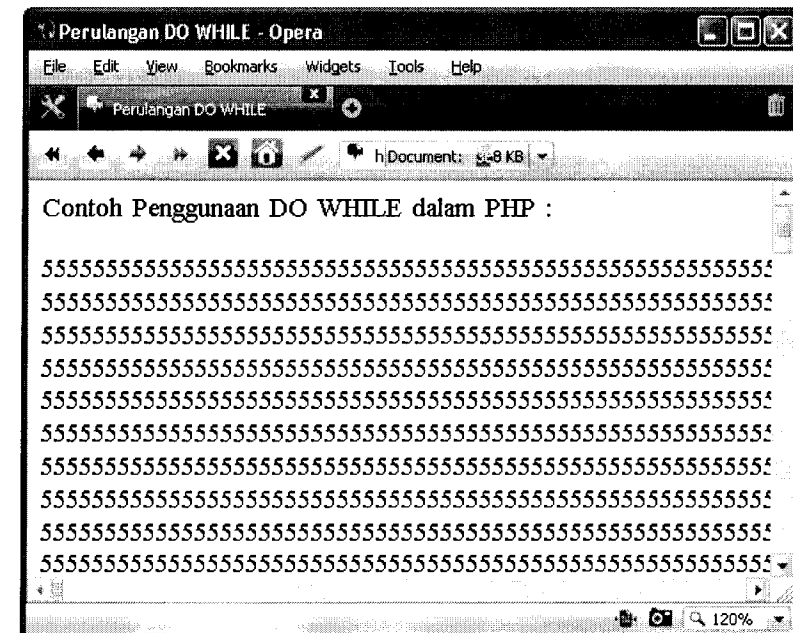
```

Contoh Penggunaan DO WHILE dalam PHP :


```

<P>
<?php
$i = 5;
do {
print $i;
} while ($i>0);
?>
</P>
</BODY>
</HTML>

```

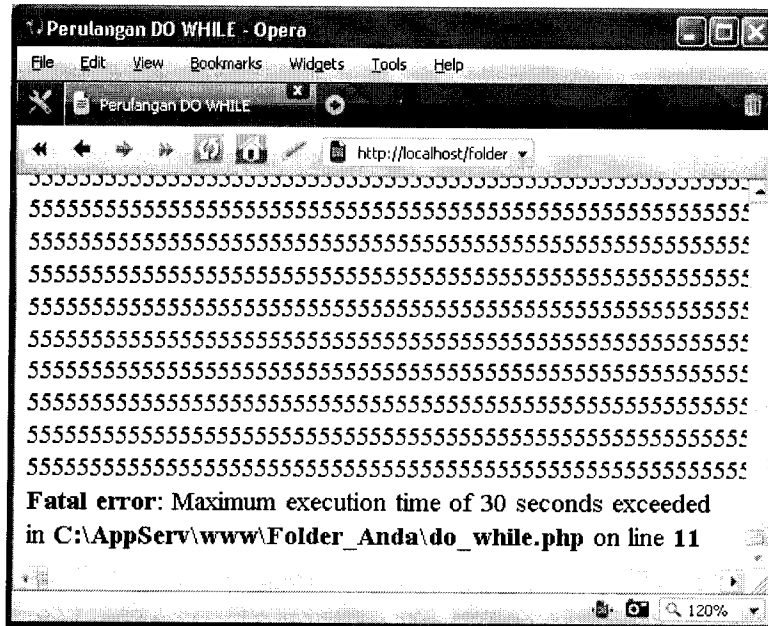


Gambar 5.26 Struktur DO WHILE Unlimited

Contoh program tersebut akan ditampilkan angka 5 berderet yang tidak terbatas. Hal tersebut terjadi karena adanya sintaks program berikut:

```
$i = 5;
do {
    print $i;
} while ($i>0);
```

Pada statement While menggunakan keterangan (\$i>0).



Gambar 5.27 Struktur DO WHILE Unlimited Error Message

Setelah 30 detik akan muncul pesan error karena server tidak mampu menampung hasil *executable* dari sintaks yang dibuat.

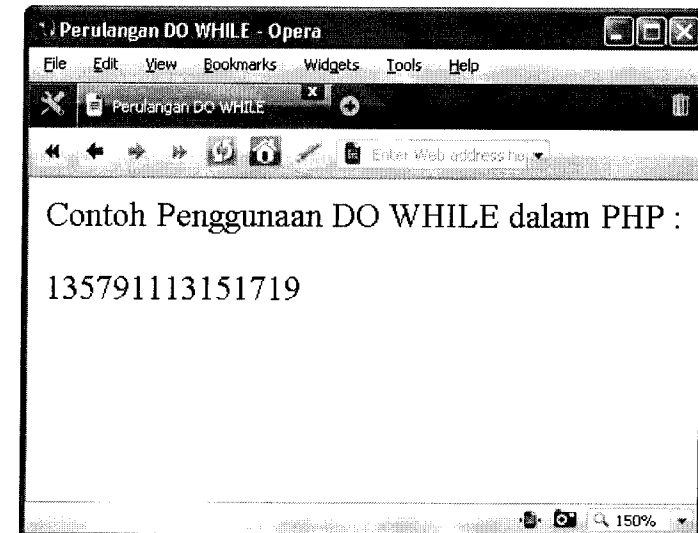
Contoh program:

```
<HTML>
<HEAD>
<TITLE> Perulangan DO WHILE </TITLE>
</HEAD>
```

<BODY>

Contoh Penggunaan DO WHILE dalam PHP :


```
<?php
$n = 1;
Do
{
    echo "$n";
    $n+=2;
}
while ($n<=20);
?>
</BODY>
</HTML>
```



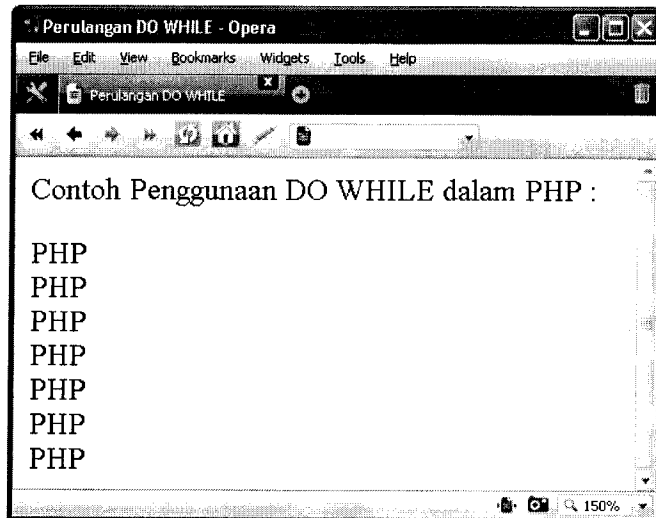
Gambar 5.28 Struktur DO WHILE Bilangan Ganjil

Contoh program:

```
<HTML>
<HEAD>
<TITLE>Perulangan DO WHILE</TITLE>
</HEAD>
<BODY>
```

Contoh Penggunaan DO WHILE dalam PHP :


```
<?php
    $php = 0;
    do {
        echo("PHP<BR>\n");
        $php++;
    } while ($php < 7);
?>
</BODY>
</HTML>
```



Gambar 5.29 Struktur DO WHILE Proses Looping PHP

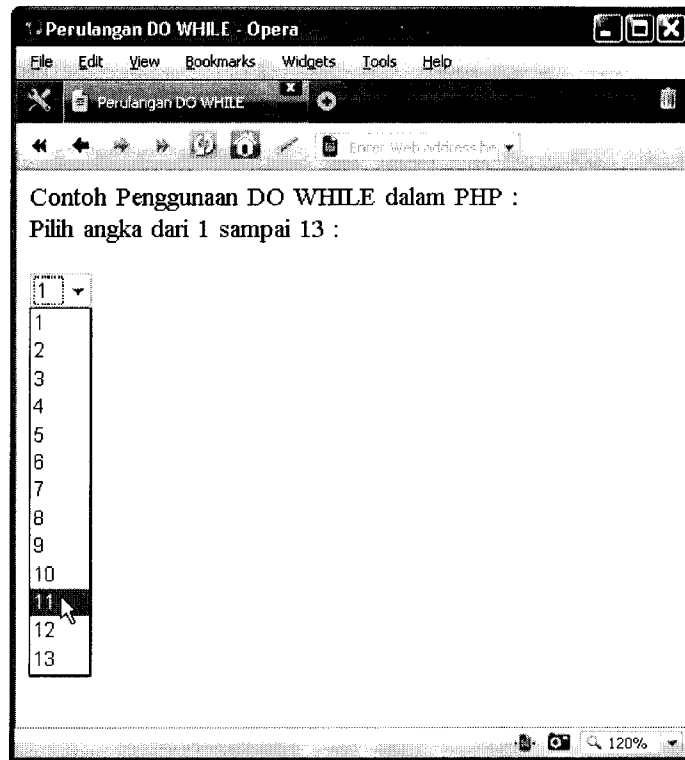
Contoh program:

```
<HTML>
<HEAD>
<TITLE>Perulangan DO WHILE</TITLE>
</HEAD>
<BODY>
```

Contoh Penggunaan DO WHILE dalam PHP :

Pilih angka dari 1 sampai 13 :

```
<P>
<?php
    $pil=13;
    echo "<select name='pilih'>\n";
        $n=1;
        do
        {
            echo("\t<option value=$n>$n</option>\n");
        }while($n++<$pil);
    echo "</select>\n";
?>
</P>
</BODY>
</HTML>
```



Gambar 5.30 Struktur DO WHILE pada Tag <OPTION VALUE>

5.2.5 BREAK

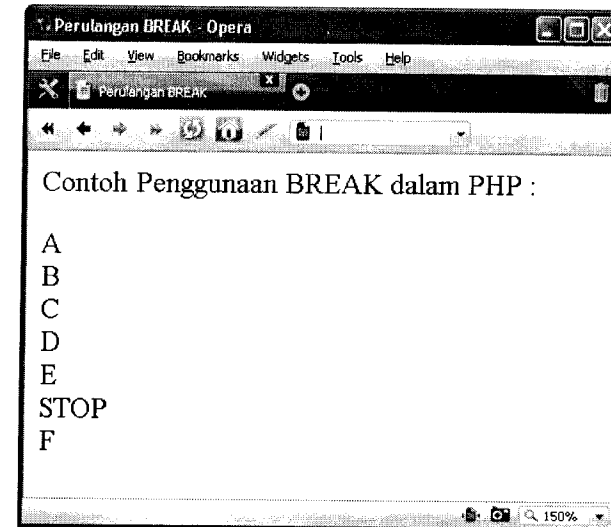
Proses looping/perulangan pada PHP memiliki statement **Break**, yang digunakan untuk menghentikan suatu proses For, While, dan Do-While. Statement break juga menerima argumen yang dapat berupa bilangan numerik.

Contoh:

```
<HTML>
<HEAD>
<TITLE> Perulangan BREAK</TITLE>
</HEAD>
<BODY>
```

Contoh Penggunaan BREAK dalam PHP :

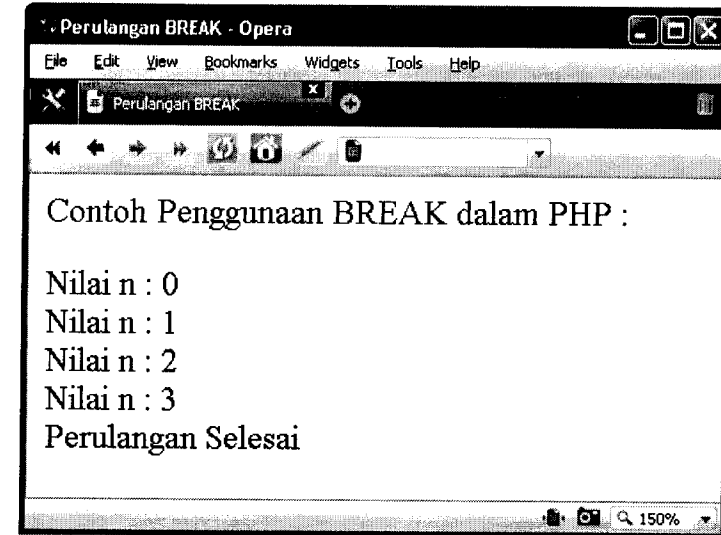

```
<P>
<?php
$huruf=array('A','B','C','D','E','BERHENTI','F');
while(list(,$val)=each($huruf))
{
    if($val == 'BERHENTI')
    {
        break;
    }
    print "$val<br>\n";
}
?>
</P>
</BODY>
</HTML>
```



Gambar 5.31 Struktur BREAK

Contoh program:

```
<HTML>
<HEAD>
<TITLE>Perulangan BREAK</TITLE>
</HEAD>
<BODY>
Contoh Penggunaan BREAK dalam PHP : <BR>
<P>
<?php
// melakukan perintah break pada $i == 4
for($n=0; $n<6; $n++)
{
if($n == 4)
{
break;
}
echo("Nilai n : $n <br>");
}
echo("Perulangan Selesai");
?>
</P>
</BODY>
</HTML>
```



Gambar 5.32 Struktur BREAK dengan IF dan FOR

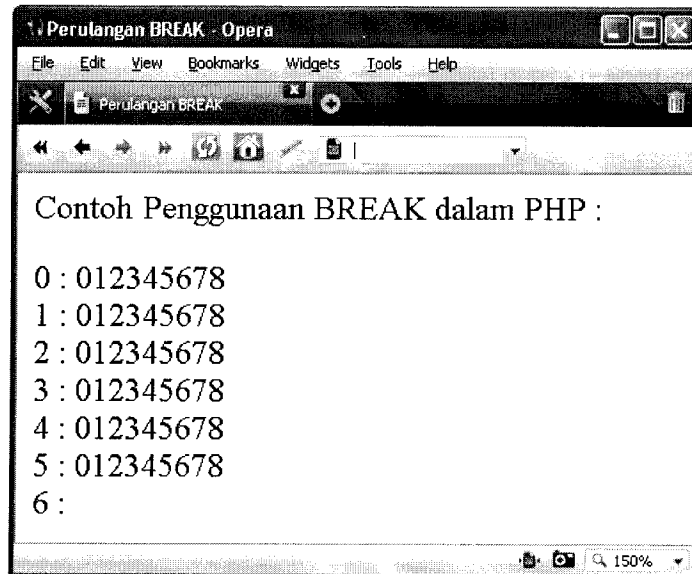
Contoh program:

```
<HTML>
<HEAD>
<TITLE> Perulangan BREAK </TITLE>
</HEAD>
<BODY>
Contoh Penggunaan BREAK dalam PHP : <BR>
<P>
<?php
for ($x = 0; $x <= 10; $x++) {
    print("$x : ");
    for ($y = 0; $y <= 8; $y++) {
        if ($x == 6)
            break 2;
        print($y);
```

```

    }
    print("<BR>\n");
}
?>
</P>
</BODY>
</HTML>

```



Gambar 5.33 Struktur BREAK

Contoh program:

```

<HTML>
<HEAD>
<TITLE> Perulangan BREAK </TITLE>
</HEAD>
<BODY>

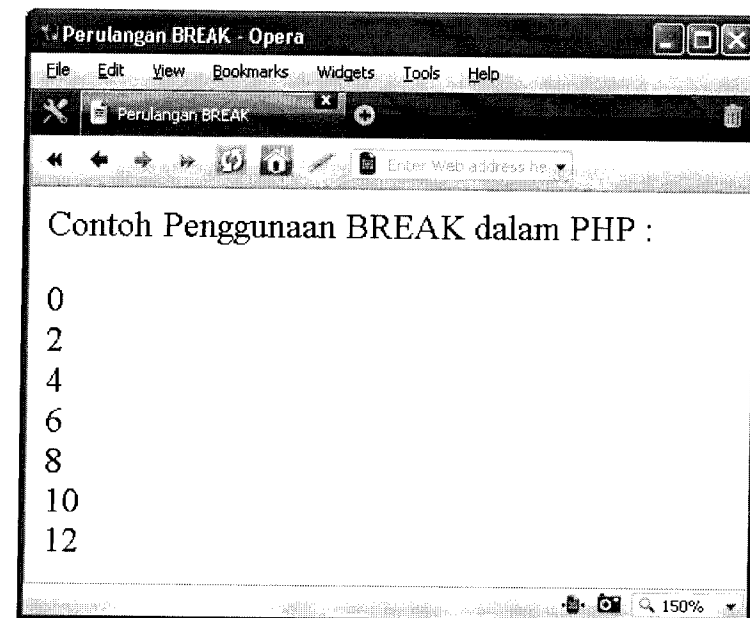
```

Contoh Penggunaan BREAK dalam PHP :


```

<P>
<?php
    for ($n = 0; $n <= 20; $n += 2)
    {
        if ($n == 14)
            break;
        print("$n<BR>\n");
    }
?>
</P>
</BODY>
</HTML>

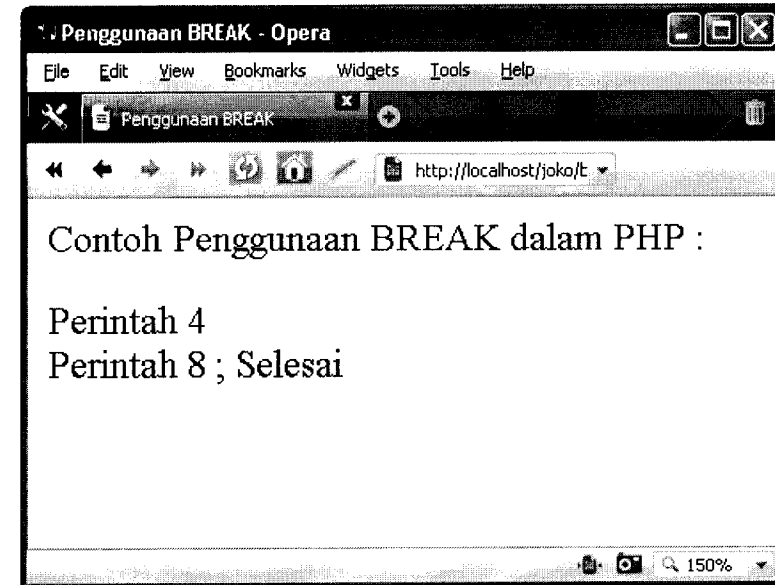
```



Gambar 5.34 Struktur BREAK Mencari Bilangan Genap

Contoh program:

```
<HTML>
<HEAD>
<TITLE> Perulangan BREAK </TITLE>
</HEAD>
<BODY>
Contoh Penggunaan BREAK dalam PHP : <BR>
<P>
<?php
$perintah=0;
while(++$perintah)
{
switch($perintah)
{
case 4 :
    echo "Perintah 4 <br>\n";
    break 1;
case 8 :
    echo "Perintah 8 ; Selesai <br>\n";
    break 2;
default;
    break;
}
}
?>
</P>
</BODY>
</HTML>
```



Gambar 5.35 Struktur BREAK dengan SWITCH

5.2.6 CONTINUE

Perintah *looping* selanjutnya adalah **Continue**, yang digunakan untuk mengarahkan atau melewati eksekusi program dalam satu putaran/iterasi ke proses perulangan selanjutnya pada *loop* sejenis.

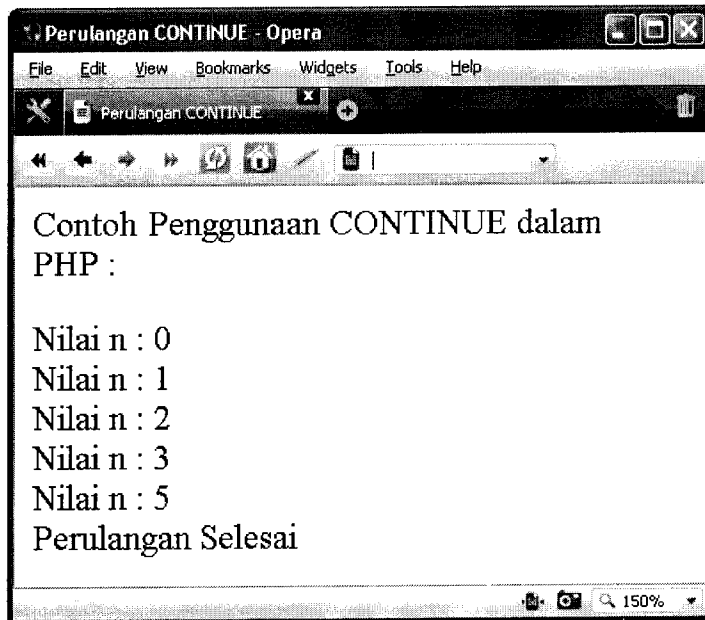
Contoh program:

```
<HTML>
<HEAD>
<TITLE> Perulangan CONTINUE </TITLE>
</HEAD>
<BODY>
Contoh Penggunaan CONTINUE dalam PHP : <BR>
<P>
<?php
// melakukan perintah continue pada $n == 4
```

```

for($n=0; $n<6; $n++)
{
if($n == 4)
{
continue;
}
echo("Nilai n : $n <br>");
}
echo("Perulangan Selesai");
?>
</P>
</BODY>
</HTML>

```



Gambar 5.36 Struktur BREAK dengan IF dan FOR

Contoh program:

```

<HTML>
<HEAD>
<TITLE> Perulangan CONTINUE </TITLE>
</HEAD>
<BODY>

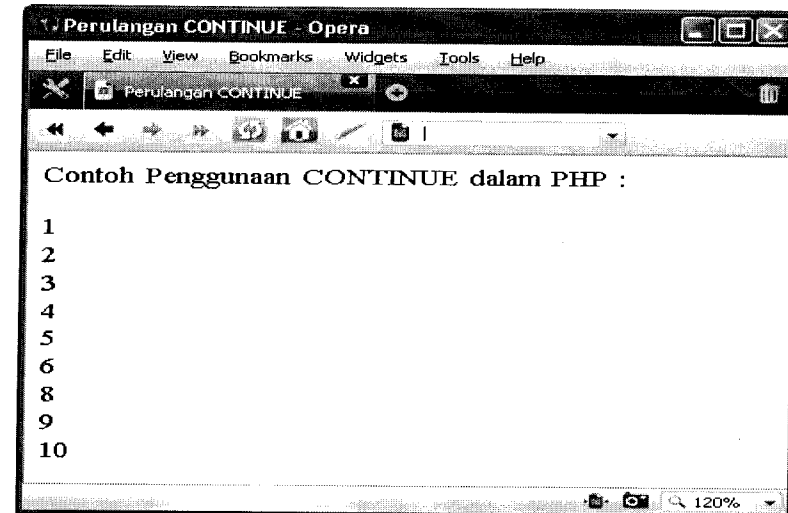
```

Contoh Penggunaan CONTINUE dalam PHP :


```

<?php
for ($n = 1; $n <= 10; $n++) {
    if ($n == 7)
        continue;
    print("$n <BR>\n");
}
?>
</BODY>
</HTML>

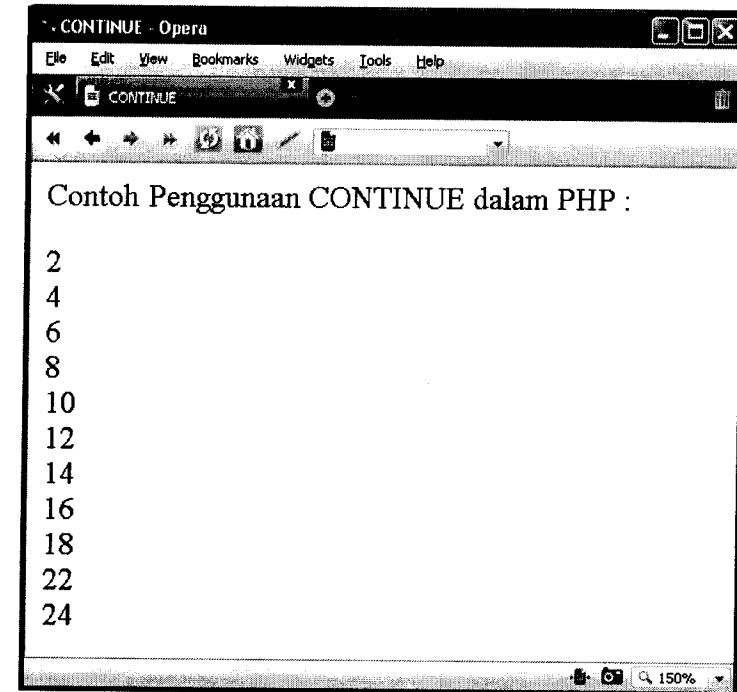
```



Gambar 5.37 Struktur CONTINUE

Contoh program:

```
<HTML>
<HEAD>
<TITLE>CONTINUE</TITLE>
</HEAD>
<BODY>
Contoh Penggunaan CONTINUE dalam PHP : <BR>
<P>
<?php
for ($angka = 2; $angka <= 25; $angka+=2) {
    if ($angka == 20)
    {
        continue;
    }
    print "$angka<br>\n";
}
echo "$angka<br>\n";
}
?>
</P>
</BODY>
</HTML>
```



Gambar 5.38 Struktur CONTINUE Bilangan Genap

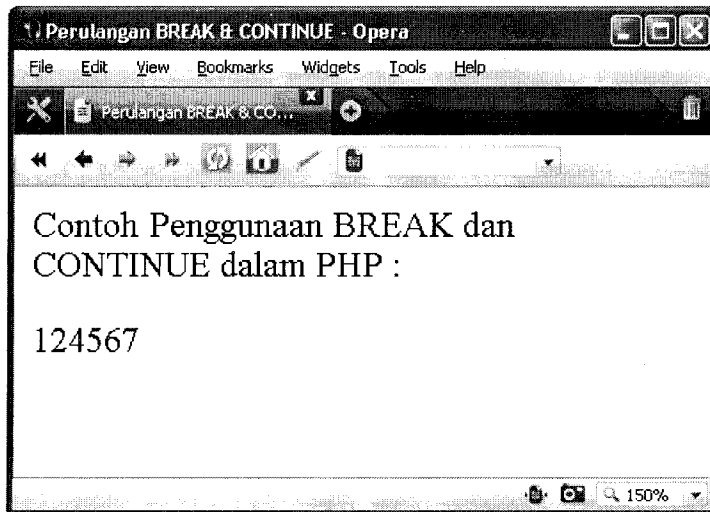
Contoh program:

```
<HTML>
<HEAD>
<TITLE> Perulangan CONTINUE </TITLE>
</HEAD>
<BODY>
Contoh Penggunaan BREAK dalam PHP : <BR>
<P>
<?php
for ($n=1; $n<18; $n++)
{
    if ($n == 3)
```

```

        continue;
    if ($n == 8)
        break;
    echo "$n";
}
?>
</P>
</BODY>
</HTML>

```



Gambar 5.39 Struktur CONTInUE dan BREAK

Dari contoh program tersebut, kita dapat menyimpulkan bahwa perintah Continue akan melanjutkan proses *looping* tanpa mengeksekusi baris perintah di bawahnya. Dengan demikian, maka perintah pada baris ke 3 - 8 akan dilewati atau 3 tidak akan dicetak, sedangkan pada perintah Break akan menyebabkan program menghentikan perulangan.

5.3 RANGKUMAN

Saat mengunjungi situs web terdapat beberapa menu yang mengharuskan Anda memasukkan pilihan. Server akan membaca perintah yang Anda berikan, kemudian PHP akan memberikan informasi yang sesuai. Kemampuan suatu program dalam mengambil keputusan inilah yang diperlukan dalam membuat sebuah website yang dinamis dan pintar. Pembuatan sebuah program yang dapat mengambil keputusan berdasarkan bermacam-macam input dari user, tergantung browser yang digunakan atau informasi yang diharapkan.

Bentuk perintah dalam PHP yang dijalankan secara urut, mulai baris pertama hingga baris terakhir, merupakan suatu struktur kontrol yang paling dasar, yang berupa urutan (sequence). Akan tetapi, perintah dalam aplikasi yang dikembangkan pada PHP tidak selalu mempunyai struktur kontrol yang berurutan. Seringkali urutan perintah tersebut tidak harus dikerjakan jika ada suatu kondisi yang tidak terpenuhi.

5.4 KUIS

1. Apakah perbedaan struktur kondisi IF, IF ELSE, dan IF THEN ELSE?
2. Sebutkan beberapa perbedaan struktur kondisi SWITCH CASE dan IF THEN ELSE!
3. Dalam PHP, FOR digunakan untuk melakukan perulangan suatu tipe data. Sebutkan tipe data yang dapat digunakan untuk proses *looping*!
4. Pernyataan DO..WHILE dan WHILE sebenarnya hampir sama. Namun ada yang membedakan dari dua struktur kondisi tersebut. Jelaskan !
5. Dalam PHP, struktur BREAK digunakan untuk menghentikan suatu proses. Jelaskan!

5.5 LATIHAN

Buatlah skrip PHP dengan ketentuan sebagai berikut:

- Buatlah skrip yang menampilkan nama, alamat, tempat tanggal lahir, dan nomor telepon dengan menggunakan perulangan sebanyak 17 kali.

- Buatlah suatu bentuk pengandaian dengan analogi: Jika saya lapar, maka saya akan makan; jika haus, maka saya akan minum; dan jika tidak keduanya, maka saya tidak lapar dan haus.

Sesi VI

Function dan Class

Durasi	2 jam
Jenis	Teori dan Praktik.
Media	PHP 5, Web Browser.
Tujuan	Setelah membaca sesi ini, siswa diharapkan: ❖ Mampu merancang situs dalam bahasa PHP dengan menggunakan dukungan Function yang tersedia dalam PHP. ❖ Dapat menggabungkan skrip PHP dengan menggunakan class yang dapat didukung dalam PHP 5.

Pada sesi sebelumnya, kita telah membahas tentang berbagai kemampuan yang terdapat pada PHP, seperti dalam bahasa pemrograman pada umumnya. Pada bab ini, kita akan mempelajari berbagai macam Fungsi dan Class yang digunakan dalam PHP 5 yang sebenarnya terdapat pada PHP 4, namun fungsi yang disediakan kurang begitu lengkap.

Fungsi dan Class merupakan pengembangan baru dari PHP 5 berdasarkan Object Oriented atau OOP. OOP singkatan dari Object Oriented Programming, yang penggunaannya akan memudahkan para programmer web. Ada beberapa istilah inti dari keistimewaan OOP tersebut, salah satunya yang cukup terkenal adalah Class, Property, Method, Inheritance, Polymorphisme, dan Encapsulation. **Class** merupakan template untuk objek yang akan dibuat.

OOP sering digunakan dalam pembuatan aplikasi secara besar -ketika penambahan fitur terbaru fungsi Encapsulation atau pembungkusan dalam *class* berperan sangat penting. Keuntungan lain adalah adanya *maintenance* ketika ditemukan bug pada kode ataupun *maintenance* jenis lain dengan mudah akan terkontrol dan menghasilkan pengkodean yang lebih efisien. Contoh nyata OOP adalah Button, turunannya seperti: Bit Button, Push Button, dan lain-lain. Semua *button* tersebut pada dasarnya merupakan bentuk lain dari *button* induk yang telah diberikan fitur tambahan dalam OOP.

OOP bukan hanya sekadar penulisan kode yang berbeda, namun lebih dari itu, OOP merupakan suatu cara pandang dalam menganalisa pemrograman dan sistem. Setiap bagian yang terdapat dalam program OOP adalah *object*. Sebuah *object* mewakili suatu bagian program yang harus diselesaikan.

Beberapa konsep dasar OOP adalah:

- Encapsulation (Class dan Object).
- Inheritance (Penurunan sifat).
- Polymorphisme.

6.1 FUNGSI

Fungsi adalah sekumpulan kode yang didefinisikan menggunakan tugas tertentu dan dapat menjadi bagian dari suatu program yang lain. Pada dasarnya, pembuatan skrip program yang memiliki fungsi sama akan sangat merepotkan seorang programmer. Hal ini sangat tidak efektif dan

tidak efisien karena ukuran file akan membesar, yang akan berdampak pada kecepatan program. Oleh karena itu, PHP telah menyediakan solusi untuk mengatasi permasalahan tersebut, yaitu dengan **FUNGSI**. Fungsi dapat dipanggil dengan menggunakan **include()**. Fungsi bekerja dengan menerima *value* dari luar -dikenal dengan nama Parameter, kemudian *value* tersebut dapat diolah dan hasilnya dapat digunakan saat di luar fungsi.

Jenis dari fungsi terbagi menjadi dua (2), yaitu:

- **Internal Function:** fungsi yang sudah tersedia pada library PHP. Anda hanya perlu mengaktifkannya ketika membutuhkan.
- **Define Function:** fungsi yang dapat dibuat sendiri.

Sintaks akan lebih mudah dibaca karena skrip akan lebih sedikit, sehingga apabila terjadi suatu kesalahan akan lebih mudah dikoreksi dan suatu perintah atau prosedur dalam file PHP dapat langsung diubah tanpa harus mengubah program utama. Itulah salah satu manfaat yang diperoleh jika Anda menggunakan fungsi.

Struktur dasar fungsi dalam PHP adalah sebagai berikut:

```
function nama_fungsi()
{
    Taruh kode program di sini;
}
```

Beberapa hal yang perlu diperhatikan saat membuat Fungsi, antara lain:

- Setiap pembuatan suatu fungsi selaluawali dengan FUNCTION.
- Penamaan suatu fungsi dapat menggunakan huruf, angka, atau keduanya. Namun demikian, penggunaan huruf tetap menjadi prioritas, sedangkan angka tidak boleh diletakkan sebelum huruf.

Contoh penamaan fungsi yang salah:

```
function 1coba ()
{
    coba;
}
```

- Penamaan fungsi tidak boleh sama dengan nama yang terdapat pada library PHP.

Contoh:

Function echo()

```
{
```

Script PHP;

```
}
```

- Tanda {...} merupakan tempat untuk menuliskan kode atau sintaks program.
- Anda dapat memanggil fungsi yang dihendaki dengan mengetikkan nama fungsinya.

Contoh program:

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>FUNCTION</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

Contoh Penggunaan FUNCTION dalam PHP :


```
<?
```

```
function umur()
```

```
{
```

```
    $age=24;
```

```
}
```

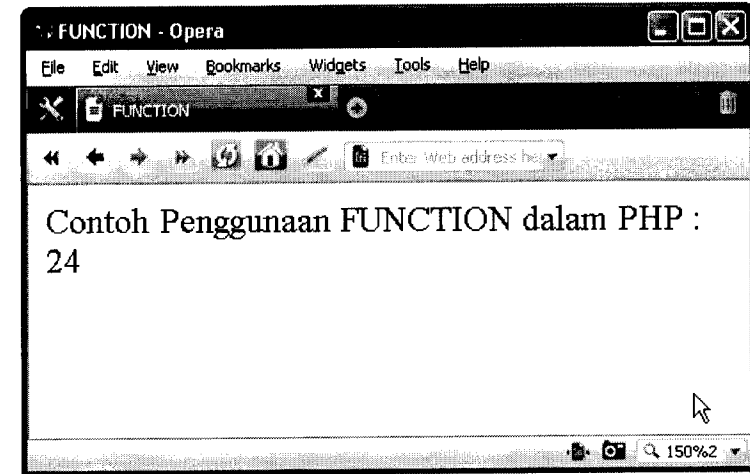
```
umur();
```

```
print (" $age");
```

```
?>
```

```
</BODY>
```

```
</HTML>
```



Gambar 6.1 Function Sederhana

Pada PHP terdapat beberapa cara dalam mendeklarasikan suatu bentuk fungsi yang terbagi menjadi 3 yaitu Built-in, User Defined Function (UDF), dan External.

6.1.1 FUNGSI BUILT-IN

Fungsi Built-in merupakan fungsi yang sudah tersedia dalam PHP dan dapat langsung digunakan, misalnya:

- **Fungsi Matematika:** sin, cos, tan, deg2rad, dan lainnya.
- **Fungsi Tanggal dan Waktu:** date, checkdate.
- **Fungsi String:** strlen, strtolower, strtoupper, substr, dan lainnya.

Contoh program:

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>FUNCTION BUILT IN</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

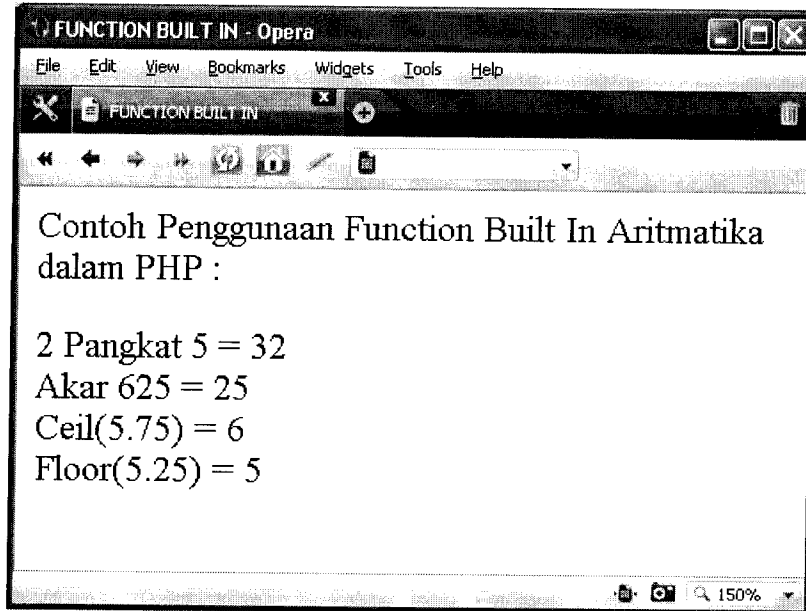
Contoh Penggunaan Function Built In Aritmatika dalam PHP :


```
<P>
```

```

<?php
$a = pow(2,5); //Fungsi perpangkatan
$b = sqrt(625); //Fungsi akar
$c = ceil(5.75); //Pembulatan keatas
$d = floor(5.25); //Pembulatan kebawah
echo "2 Pangkat 5 = $a <br>";
echo "Akar 625 = $b <br>";
echo "Ceil(5.75) = $c <br>";
echo "Floor(5.25) = $d <br>";
?>
</P>
</BODY>
</HTML>

```



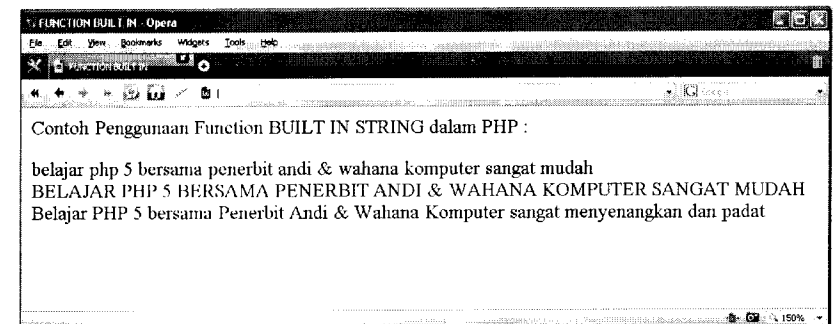
Gambar 6.2 Function Built-in Matematika

Contoh program:

```

<HTML>
<HEAD>
<TITLE> FUNCTION </TITLE>
</HEAD>
<BODY>
Contoh Penggunaan Function BUILT IN STRING dalam PHP : <BR>
<P>
<?php
$str = "Belajar PHP 5 bersama Penerbit Andi & Wahana Komputer sangat Mudah";
echo strtolower($str); //Ubah huruf ke kecil semua
echo "<br>";
echo strtoupper($str); //Ubah huruf ke besar semua
echo "<br>";
echo str_replace("Mudah","menyenangkan dan padat",$str); //Mengganti string
?>
</P>
</BODY>
</HTML>

```



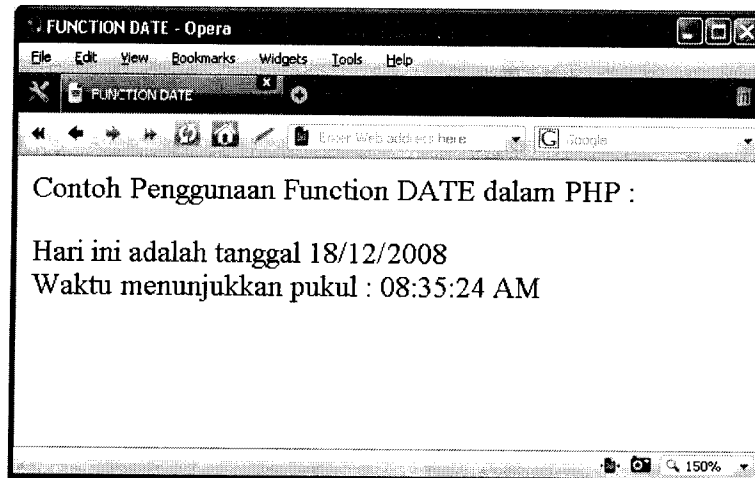
Gambar 6.3 Penggunaan Function BUILT IN STRING

Contoh program:

```
<HTML>
<HEAD>
<TITLE> FUNCTION DATE</TITLE>
</HEAD>
<BODY>
```

Contoh Penggunaan Function DATE dalam PHP :


```
<P>
<?php
$tgl = date("d/m/Y");
echo "Hari ini adalah tanggal $tgl <br>";
$waktu = date("h:i:s A"); //A menunjukkan AM atau PM
echo "Waktu menunjukkan pukul : $waktu";
?>
</P>
</BODY>
</HTML>
```



Gambar 6.4 Penggunaan Function DATE

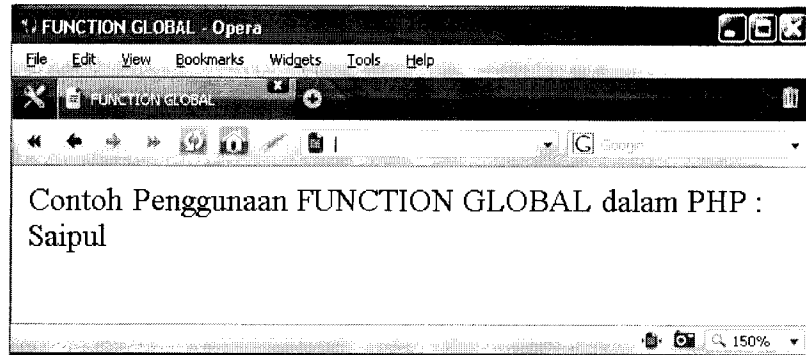
Berbeda dengan Tipe Data, Function masih tetap menggunakan variabel `date("d/m/Y")`;; namun dalam Function dapat ditambahkan `date("h:i:s A")`; yang bertujuan sebagai penanda waktu AM/PM.

Contoh program:

```
<HTML>
<HEAD>
<TITLE>FUNCTION GLOBAL</TITLE>
</HEAD>
<BODY>
```

Contoh Penggunaan FUNCTION GLOBAL dalam PHP:

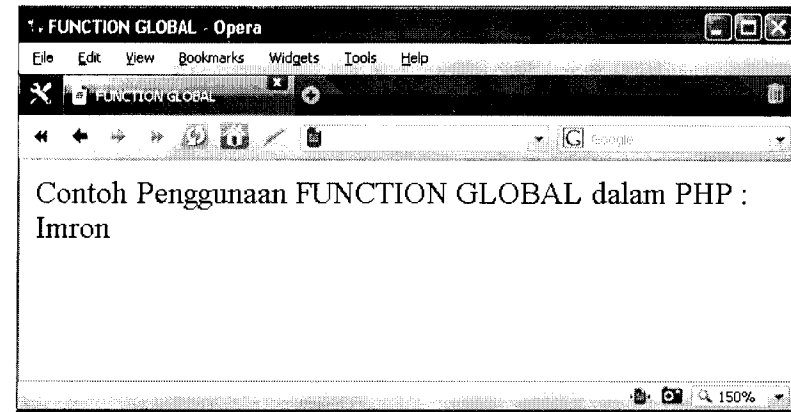

```
<?php
function alih()
{
    global $siswa;
    $siswa = "Saipul";
}
$siswa = " Imron ";
alih();
print("$siswa");
?>
</BODY>
</HTML>
```



Gambar 6.5 Penggunaan Function GLOBAL Alih Nama

Contoh program:

```
<HTML>
<HEAD>
<TITLE>FUNCTION GLOBAL</TITLE>
</HEAD>
<BODY>
Contoh Penggunaan FUNCTION GLOBAL dalam PHP : <BR>
<?php
function alih() {
    $GLOBALS["siswa"] = "Imron";
}
$siswa = "Saipul";
alih();
echo("$siswa");
?>
</BODY>
</HTML>
```



Gambar 6.6 Penggunaan Function GLOBAL

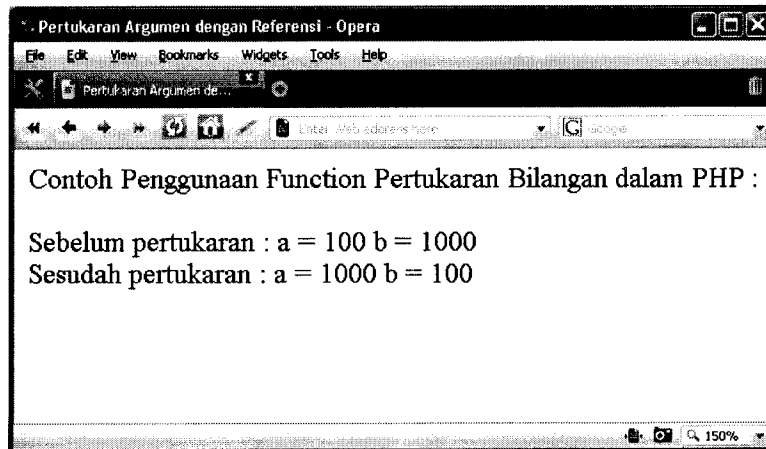
Contoh program:

```
<HTML>
<HEAD>
<TITLE>Pelewatan Argumen dengan Referensi</TITLE>
</HEAD>
<BODY>
```

Contoh Penggunaan Function Pertukaran Bilangan dalam PHP :


```
<?php
function tukar(&$x, &$y) {
    $tmp = $x;
    $x = $y;
    $y = $tmp;
}
$a = 100;
$b = 1000;
print("Sebelum pertukaran : ");
printf("a = %d b = %d <BR>\n", $a, $b);
tukar($a, $b);
```

```
print("Sesudah pertukaran : ");
printf("a = %d b = %d <BR>\n", $a, $b);
?>
</BODY>
</HTML>
```

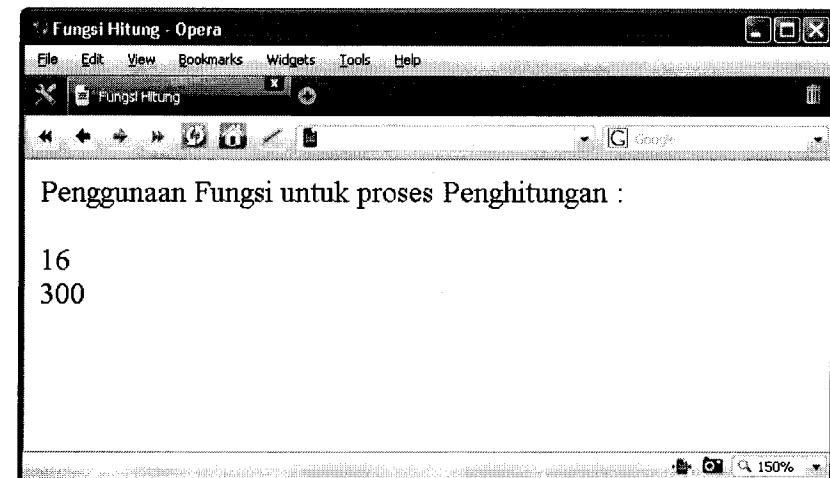


Gambar 6.7 Penggunaan Function Pertukaran

Contoh program:

```
<HTML>
<HEAD>
<TITLE>Fungsi Hitung</TITLE>
</HEAD>
<BODY>
Penggunaan Fungsi untuk proses Penghitungan : <BR> <BR>
<?php
function hasil($a, $b)
{
    $hitung = $a + $b;
    return $hitung;
```

```
}
$c = hasil(10, 6);
echo($c);
echo("<BR>\n");
echo(hasil(200, 100));
?>
</BODY>
</HTML>
```



Gambar 6.8 Penggunaan Function untuk Menghitung Suatu Bilangan

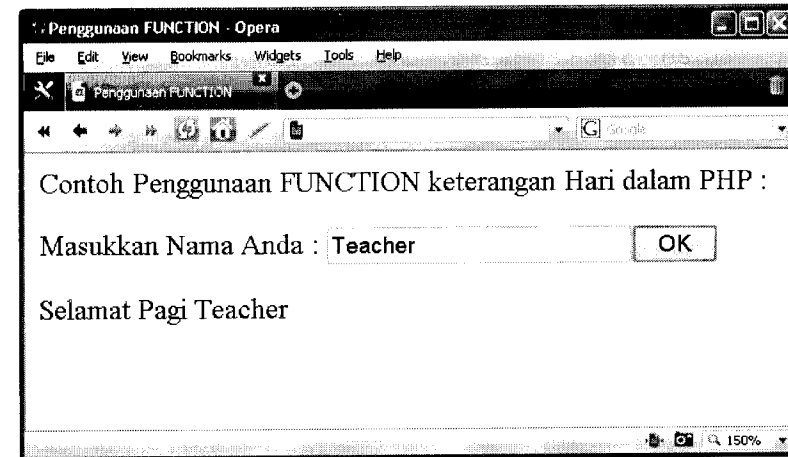
Contoh program:

```
<HTML>
<HEAD>
<TITLE> Penggunaan FUNCTION </TITLE>
</HEAD>
<BODY>
Contoh Penggunaan FUNCTION keterangan Hari dalam PHP : <BR>
<?php
```

```

echo "<form action='$PHP_SELF?action=1' method='post'>";
echo "Masukkan Nama Anda : <input type='text' name='nama'
      size=25>";
echo "<input type='submit' value=' OK '></form>";
      if($action==1)
{
    greet($nama);
}
function greet($str)
{
    $date = date(G);
    if($date<11 && $date>=0)print"Selamat Pagi $str";
    else if($date>=11 && $date<15)print "Selamat Siang $str";
    else if($date>=15 && $date<18)print "Selamat Sore $str";
    else print"Selamat Malam $str";
}
?>
</BODY>
</HTML>

```



Gambar 6.9 Penggunaan Function untuk Mencari Keterangan Hari

6.2 CLASS DALAM PHP

Pada awalnya, PHP diciptakan oleh Rasmus Lerdorf beserta tim, PHP hanya mendukung konsep pemrograman terstruktur. Sampai versi 4, PHP masih berkonsep terstruktur hingga akhirnya diciptakan PHP versi 5, yang telah mendukung konsep orientasi objek (OOP). Apakah kelebihan/keuntungan dari konsep orientasi objek dibandingkan dengan konsep prosedural? Ternyata, cukup banyak keuntungannya yang bisa didapatkan. Konsep orientasi objek mendukung pewarisan dan memiliki kemampuan untuk menggunakan nama variabel yang sama secara berulang-ulang. Alasan tersebut menjadikan konsep OOP sangat baik digunakan dalam pembuatan proyek pembangunan web berskala besar.

Class adalah sekumpulan dari variabel/property dan fungsi/method yang dapat mengolah variabel-variabel ini. Class sering disebut sebagai **object**. Penggunaan object pada proses pembuatan program menjadi lebih cepat, karena object-object yang sudah ada dapat dipakai untuk membuat objek yang lain. Class juga dapat diartikan sebagai suatu kumpulan variabel dan fungsi yang bekerja dengan suatu variabel. Sebagai contoh, class **manusia**. Class **manusia** setidaknya bisa memiliki bagian, seperti: kepala, tangan, kaki, telinga, hidung, mulut, dan lain-lain. Class **orang** juga setidaknya harus bisa berjalan, berlari, melompat, bicara, dan lain-lain. Dengan adanya suatu konsep pemrograman berorientasi objek, para programmer PHP akan lebih mudah berbagi

teknik pemrograman. Anda dapat membuat suatu class dan programmer dapat dengan mudah menggunakannya tanpa harus lebih tahu bagaimana proses jalannya class tersebut. Di internet sudah banyak bertebaran berbagai macam class dengan fungsinya masing-masing yang tersedia secara gratis dan dapat dimodifikasi secara bebas.

6.2.1 CLASS

Konsep dasar class:

```
<?php
class daftarMenu {
    // properties
    // methods
}
?>
```

Pada contoh tersebut, sebuah kelas dengan nama **daftarMenu** sudah berhasil dibuat, tetapi belum dapat dilakukan apa-apa pada kelas tersebut karena Anda belum menambahkan properti (properties) atau metode (methods) di dalamnya.

```
<?php
class daftarMenu {
    function Nasi() {
        echo "Nasi Goreng";
    }
}
?>
```

Sekarang, Anda telah meletakkan sebuah fungsi **Nasi()** pada kelas menu. Fungsi pada kelas disebut juga metode (method). Kelas tersebut belum dapat digunakan, karena penggunaan kelas tersebut beserta metode dan propertinya membutuhkan instance dari kelas tersebut dengan perintah **new**. Dengan mendeklarasikan sebuah *instance*, maka Anda dapat menciptakan sebuah objek (object).

```
<?php
```

```
class daftarMenu {
    function Nasi() {
        echo "Nasi Goreng";
    }
}

$menu = new daftarMenu();
$menu->Nasi(); // Hasil: Nasi Goreng
?>
```

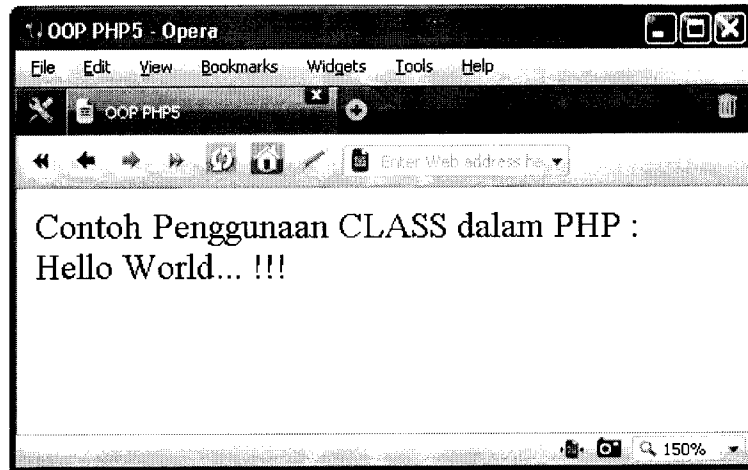
Contoh program:

```
<HTML>
<HEAD>
<TITLE>OOP PHP5 </TITLE>
</HEAD>
<BODY>
```

Contoh Penggunaan CLASS dalam PHP :


```
<?php
class cetak {
    function hallo () {
        print "Hello World... !!!";
    }
}

$cetak = new cetak();
$cetak->hallo ();
?>
</BODY>
</HTML>
```



Gambar 6.10 Class Sederhana

6.2.2 OBJECT

Contoh skrip tersebut menggunakan perintah **new** untuk menciptakan sebuah *instance* dari kelas **daftarMenu** dan telah ditetapkan (assigned) pada variabel **\$menu**, kemudian memanggil metode **Nasi()** dengan objek **\$menu**. Anda perlu memperhatikan perbedaan antara kelas dengan objek. Kelas, pada contoh program adalah **daftarMenu()**, dan **\$menu** adalah sebuah objek (object). Kelas hanya dapat dibuat satu, misalnya ketika akan membuat dua kelas dengan nama yang sama, yaitu **daftarMenu**, maka akan terjadi kesalahan (error). Meski demikian, Anda dapat membuat beberapa *instance* menggunakan satu kelas yang sama.

```
<?php
class daftarMenu {
    function Nasi() {
        echo "Nasi Goreng";
    }
}

$menu = new daftarMenu();
$menu->Nasi(); // Hasil: Nasi Goreng
```

```
$menu2 = new daftarMenu();
$menu2->Nasi(); // Hasil: Nasi Goreng

?>
```

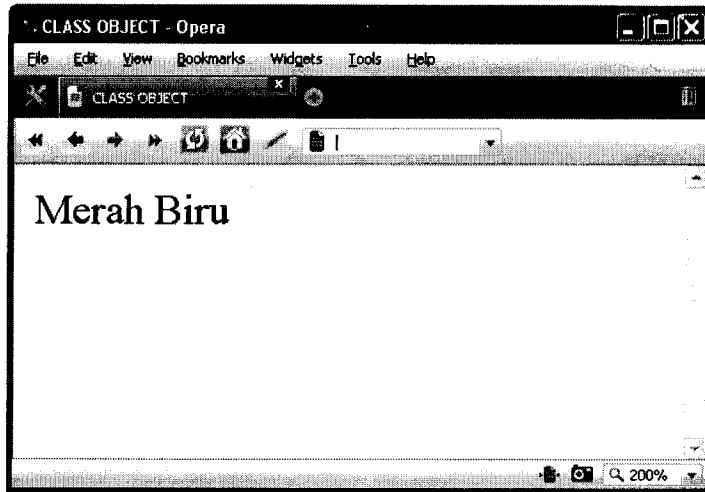
Pada contoh program tersebut, Anda dapat melihat bahwa terdapat dua *instance* yang dibuat dari kelas **daftarMenu**, meskipun Anda telah mempunyai 2 objek, yaitu: **\$menu** dan **\$menu2**. Masing-masing *instance* akan memanggil metode **Nasi()**.

Contoh program:

```
<HTML>
<HEAD>
<TITLE>CLASS OBJECT</TITLE>
</HEAD>
<BODY>
<?
class MyColorEnumClass {
    const RED = "Merah";
    const GREEN = "Hijau";
    const BLUE = "Biru";
    function printBiru()
    {
        print self::BLUE;
    }
}

echo MyColorEnumClass::RED."\n";
$obj = new MyColorEnumClass();
$obj->printBiru();

?>
</BODY>
</HTML>
```



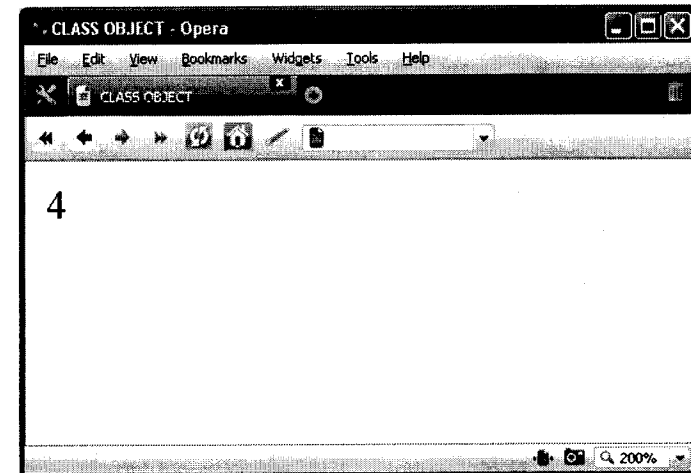
Gambar 6.11 Class Object

Contoh program:

```
<HTML>
<HEAD>
<TITLE>CLASS OBJECT</TITLE>
</HEAD>
<BODY>
<?
class MyClass {
public $var = 2;
}
$obj1 = new MyClass();
$obj2 = $obj1;
$obj2->var = 4;
print $obj1->var;
?>
```

```
</BODY>
```

```
</HTML>
```



Gambar 6.12 Cloning Object

6.2.3 PROPERTIES

Sebuah kelas dapat memiliki berbagai variabel yang disebut properti (properties) atau bisa disebut juga *attributes*. Tipe-tipe dari properti tersebut adalah: public, private, dan protected.

```
<?php
class daftarMenu {
    public $rasa;
    function Nasi() {
        echo "Nasi Goreng" . " " . $this->rasa;
    }
}
$menu = new daftarMenu();
$menu->rasa = "Sangat Pedas";
```

```
$Menu->Nasi(); // Hasil: Nasi Goreng Sangat Pedas
```

```
?>
```

Dalam contoh program tersebut, Anda dapat melihat tambahan sebuah baris, yaitu “**public \$rasa;**“, yang disebut juga *access modifiers*. Dengan kata lain, Anda telah menambahkan sebuah properti public dengan nama **\$rasa** pada kelas yang telah dibuat sebelumnya. Pada metode **Nasi()** juga sudah ditambahkan properti **\$rasa**, yang akan menampilkan pula variabel properti **\$rasa** dalam browser. Setelah itu, Anda dapat melihat penggunaan **\$this->rasa** untuk memanggil properti **\$rasa**. Pseudo-variabel **\$this** adalah sebuah variabel yang sudah tersedia ketika suatu metode dipanggil oleh sebuah objek. **\$this** adalah sebuah acuan yang dapat digunakan untuk memanggil sebuah metode atau properti pada sebuah kelas.

Awal pembuatan program memang akan terasa sulit, berbeda dengan konsep terstruktur. Namun demikian, Anda disarankan untuk menggunakan konsep OOP dengan variabel yang sangat banyak dalam pembangunan proyek besar.

6.2.4 INHERITANCE

Salah satu kelebihan OOP adalah memiliki CLASS, sedangkan salah satu kelebihan menggunakan sebuah kelas adalah Anda dapat membuat kelas *inherits*. Pewarisan (inheritance) dilakukan dengan memperpanjang (*extends*) kelas yang sudah ada dengan cara membuat kelas baru yang mewarisi (*inherits*) semua properti dan metode yang terdapat pada kelas aslinya (*parent class*).

Pewarisan (dalam OOP disebut **Inheritance**) berfungsi untuk menggunakan kembali metode dari suatu kelas yang sebelumnya telah dibuat. Sebagai contoh, Anda menggunakan method **Joko** pada class **X**. Dengan demikian, Anda dapat memanggil method **Joko** tersebut pada class **B** dengan cara memanggilnya melalui class **X**, tanpa harus membuat ulang method **Joko** tersebut. Inheritance terbagi menjadi 2 jenis, yaitu: **Single Inheritance** (class anak mendapat turunan langsung dari class induknya) dan **Multiple Inheritance** (class anak mendapat turunan dari class ibu dan ayahnya).

Contoh program:

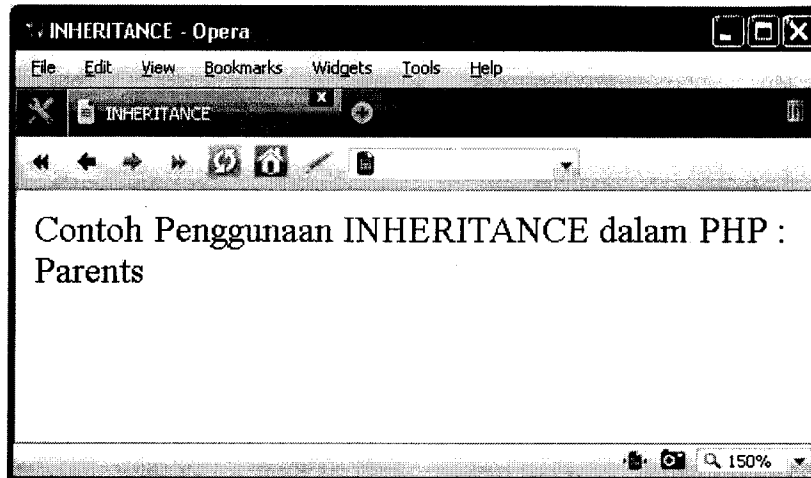
```
<HTML>
<HEAD>
<TITLE>INHERITANCE</TITLE>
</HEAD>
<BODY>
```

Contoh Penggunaan INHERITANCE dalam PHP :


```
<?php
class Parents {
var $Parents = "Parents";
function Parents () { #baris ini merupakan konstruktor#
$this->Parents;
}
}

class Child extends Parents{
function Child () {
print $this->Parents;
}
}

$Child = new Child ();
?>
</BODY>
</HTML>
```



Gambar 6.13 Inheritance Parents dan Child

Class Parents dapat divariabelkan kata "Parents", lalu kata "Parents" tersebut diwariskan pada class Child, sehingga ketika hendak mencetak class Child, Anda dapat memanggil variabel Parents dari class Parents.

6.2.5 POLYMORPHISM

Polymorphism (dapat memiliki berbagai bentuk), yaitu sebagai object dari class-nya sendiri atau dapat sebagai object dari superclass-nya. Polymorphism memiliki 2 bentuk, yaitu: **Overloading** dan **Overriding**.

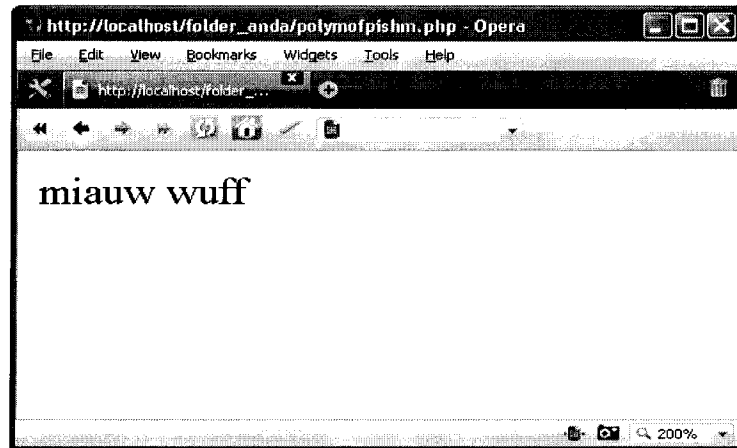
Contoh program:

```
<HTML>
<HEAD>
<TITLE> Polymorphism</TITLE>
</HEAD>
<BODY>
<?
class Animal {
function makeSound()
{
```

```
print "Error: metode ini harus diterapkan kembali pada turunan /child";
    }
}
class Cat extends Animal {
function makeSound()
{
    print "miauw";
}
}
class Dog extends Animal {
function makeSound()
{
    print "wuff";
}
}
function printTheRightSound($obj)
{
    if ($obj instanceof Animal) {
        $obj->makeSound();
    } else {
        print "Error: Passed object yang salah";
    }
    print "\n";
}
printTheRightSound(new Cat());
printTheRightSound(new Dog());
?>
```

```
</BODY>
```

```
</HTML>
```



Gambar 6.14 Class Polymorphism

Overloading

Penggunaan satu nama untuk beberapa metode dengan parameter yang berbeda.

Contoh program:

```
<?php
class metode
{
    //metode query dengan satu argumen
    public function query($query)
    {
        echo '<p>';
        echo 'Anda memanggil query($query, $connection)<br />';
        echo 'Query Anda adalah : '.$query;
        echo '</p>';
    }
}
```

```
//metode query dengan 2 argumen
```

```
public function query($query, $connection)
```

```
{
```

```
    echo '<p>';
```

```
    echo 'Anda memanggil query($query, $connection)<br />';
```

```
    echo 'Query Anda adalah : '.$query;
```

```
    echo '<br />';
```

```
    echo 'Koneksi Anda adalah : '.$connection;
```

```
    echo '</p>';
```

```
}
```

```
}
```

```
echo '<h2>Method Overloading</h2>';
```

```
$metode = new metode();
```

```
$metode->query("SELECT * FROM coba");
```

```
$metode->query("SELECT * FROM coba", "MYSQL OBJECT");
```

```
?>
```

Untuk menjalankan skrip tersebut, Anda harus membuat database dengan menggunakan MySQL terlebih dahulu dengan nama tabel **coba**.

Overriding

Mendeklarasikan metode dengan nama dan parameter yang sama dengan method pada *superclass*.

Contoh program:

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Overriding</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

Contoh Penggunaan Overriding dalam PHP :


```

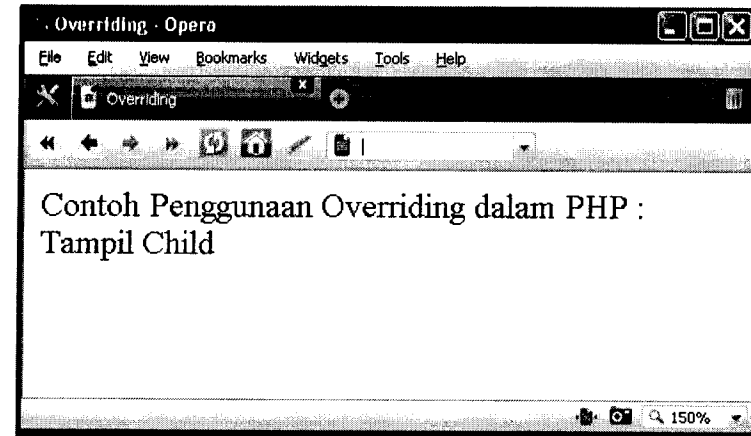
<?php
class Parents
{
    var $tampil = "Tampil Parents";
    function tampil ()
    {
        print $this->tampil;
    }
}

class Child extends Parents
{
    var $tampil = "Tampil Child";
    function tampil ()
    {
        print $this->tampil;
    }
}

$cetak = new Child ();
$cetak->tampil();
?>

</BODY>
</HTML>

```



Gambar 6.15 Polymorphisme Overriding

Anda dapat melihat tampilan kata "Tampil Child" pada contoh program yang muncul di browser. Seharusnya, pewarisan kelas Parents harus menampilkan "Tampil Parents " karena pada class Child telah terjadi *overriding* pada method kata. Oleh karenanya, tampilan yang muncul adalah "Tampil Child", bukan "Tampil Parents".

6.2.6 CONSTRUCTOR

Constructor adalah sebuah method atau behavior yang dijalankan otomatis ketika sebuah object dibuat dari sebuah class. Constructor merupakan bagian dari pemrograman berorientasi objek (OOP), berupa fungsi anggota yang mempunyai nama yang sama dengan nama class.

Kegunaannya:

- Mengalokasikan ruang bagi sebuah objek.
- Memberikan nilai awal.
- Membentuk tugas-tugas umum lainnya.

Bisa saja, suatu class tidak memiliki konstruktor atau bahkan memiliki lebih dari satu konstruktor.

Pada umumnya, konstruktor dapat dibedakan menjadi 2 jenis, yaitu:

■ Default Constructor

Jenis konstruktor ini menginisialisasikan objek dengan nilai-nilai *default* yang ditentukan oleh perancang class. Di dalam deklarasi class, konstruktor ini tidak memiliki parameter formal.

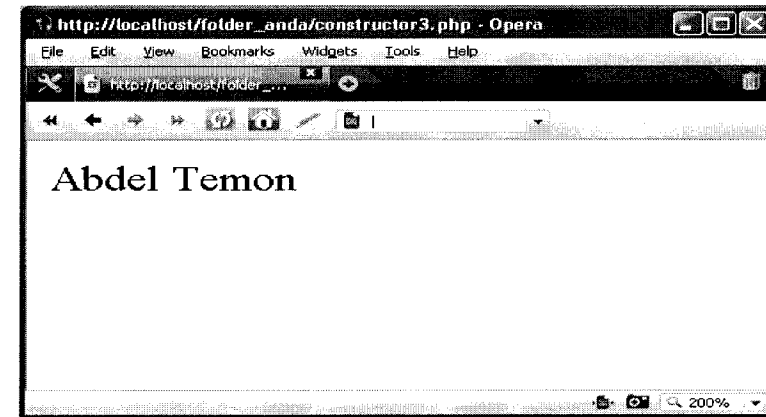
■ User-defined Constructor

Jenis konstruktor ini menginisialisasi objek dengan nilai yang diberikan oleh pemakai class pada saat objek diciptakan. Dalam deklarasi class, konstruktor ini memiliki satu atau lebih parameter formal.

Contoh Program:

```
<HTML>
<HEAD>
<TITLE> Constructors </TITLE>
</HEAD>
<BODY>
<?
class Siswa {
private $nama;
function __construct($nama)
{
$this->nama = $nama;
}
function getNama()
{
return $this->nama;
}
};
?>
<?>
```

```
$Abdel = new Siswa("Abdel");
$Temon = new Siswa("Temon");
echo $Abdel->getNama()."\n";
echo $Temon->getNama();
?>
</BODY>
</HTML>
```



Gambar 6.16 Constructor

6.2.7 DESTRUCTOR

Destructor adalah sebuah metode yang berfungsi untuk menghapus objek ketika program berakhir. Anda dapat menuliskan destruktur secara eksplisit dengan menambahkan metode dengan **__destruct()** pada sebuah class, metode tersebut akan otomatis dijalankan oleh php pada akhir eksekusi skrip.

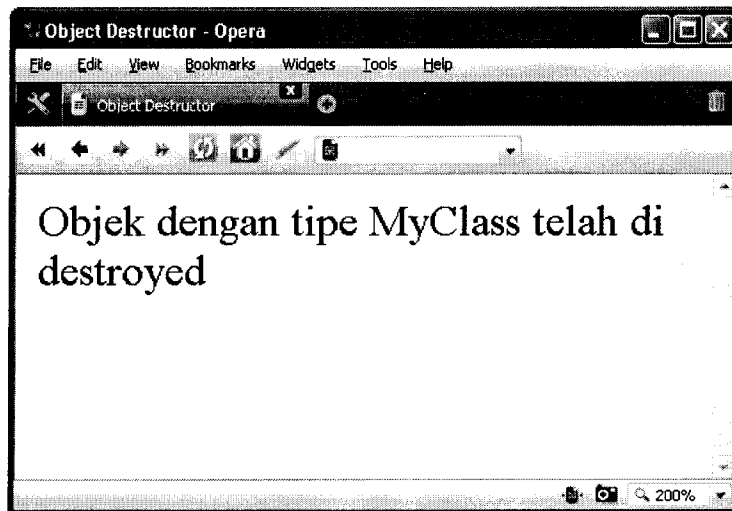
Contoh program:

```
<HTML>
<HEAD>
<TITLE>Object Destructor</TITLE>
</HEAD>
```

```

<?
class MyClass {
function __destruct()
{
echo "Objek dengan tipe MyClass telah di destroyed\n";
}
};
$obj = new MyClass();
$obj = NULL;
?>
</BODY>
</HTML>

```



Gambar 6.17 Destructor

Destructor merupakan kebalikan dari metode Constructor. Berdasarkan contoh program tersebut, jenis class yang digunakan (MyClass) telah ditentukan sebelumnya, kemudian menggunakan `function__destruct()`.

```
class MyClass {
```

```

function __destruct()
{
echo "Objek dengan tipe MyClass telah di destroyed\n";
}
};
$obj = new MyClass();
$obj = NULL;

```

Object yang bernilai NULL akan tetap ditampilkan, padahal `$obj=NULL`.

6.3 RANGKUMAN

Pada awalnya, kelahiran PHP hanya terdiri dari sekumpulan skrip sederhana. Dalam perkembangannya, kebutuhan terhadap pemrograman berbasis objek membuat PHP harus bermigrasi dari pemrograman terstruktur. Pada versi ke 4, PHP mengumumkan bahwa bahasa pemrograman tersebut sudah berbasis objek. Fitur-fitur yang dibawa membuat PHP semakin cepat dan lebih efisien. Pemrograman berbasis objek atau Object-Oriented Programming (OOP) merupakan suatu pendekatan pemrograman, yang dalam penerapannya menggunakan object dan class. Dalam perkembangannya, hampir semua situs di internet sekarang sudah menerapkan konsep OOP secara tidak langsung.

Pemrograman yang saat ini banyak digunakan dalam penerapan konsep OOP adalah Java dan C++. Konsep OOP bukan sekadar bagaimana cara penulisan yang berbeda, namun lebih dari itu, OOP merupakan suatu cara pandang dalam menganalisa sistem dan permasalahan suatu pemrograman. Setiap bagian dari program OOP dianggap sebagai object. Sebuah object dapat mewakili suatu bagian program yang akan diselesaikan; PHP 5 sudah mendukung beberapa konsep OOP.

6.4 KUIS

1. Jelaskan pengertian OOP! Mulai pada versi berapakah PHP menggunakan OOP?
2. Sebutkan beberapa keistimewaan penggunaan OOP!

3. Apakah yang dimaksud dengan Fungsi (Function) pada PHP? Sebutkan beberapa hal yang harus diperhatikan dalam penggunaan Fungsi!
4. Jelaskan penggunaan Class pada PHP!
5. Sebutkan dua jenis Polymorphism pada Class dan jelaskan kegunaan masing-masing!

6.5 LATIHAN

Buatlah skrip PHP dengan menggunakan Function yang menampilkan keterangan tentang data diri Anda, dengan ketentuan sebagai berikut:

- Nama : *Nama Anda*
- Tanggal lahir
- Alamat
- Email

Simpan *project* tersebut dengan nama **Function.php**, kemudian jalankan pada browser.

Sesi VII

Aplikasi Web dengan PHP

Durasi	2 jam
Jenis	Teori dan Praktik.
Media	PHP 5, Web Browser, Microsoft Word.
Tujuan	Setelah membaca sesi ini, siswa diharapkan dapat: ❖ Menerapkan PHP dalam suatu website. ❖ Contoh sintaks PHP yang sering digunakan. ❖ Mengembangkan web dinamis dengan PHP.

Pembahasan PHP dimulai dengan sejarah dan perkembangannya, bagaimana bentuk dan susunan bahasanya, serta menjelaskan fasilitas dan beberapa fitur yang tersedia di PHP. Pembahasan struktur pemrograman PHP sengaja dipisahkan untuk mempermudah pemahaman materi. Untuk memperjelas pembahasan mengenai PHP, pada bagian berikutnya akan dijelaskan penerapan PHP melalui sebuah contoh aplikasi web.

7.1 MULTI LANGUAGE SITE

Sebuah situs biasanya memiliki fasilitas pilihan bahasa untuk mempermudah para pengunjung situs dalam menelusuri informasi yang tersedia di situs tersebut. Selanjutnya, Anda akan belajar tentang pembuatan *multi-language* pada suatu website.

Contoh program:

Bahasa.php

```
<HTML>
<HEAD>
<TITLE>Alih Bahasa</TITLE>
</HEAD>
<BODY>
<?php
    session_start();
    $_SESSION['bhs'] = $a;
?>
<script language="JavaScript" type="text/JavaScript">
<!--
function MM_jumpMenu(targ,selObj,restore){ //v3.0
    eval(targ+"."+location="" +selObj.options[selObj.selectedIndex].value+"");
    if (restore) selObj.selectedIndex=0;
}
//-->
```

```
</script>
<form name="form" method="post" action="?bhs=">
    Pilih Bahasa :
    <select name="menu" onChange="MM_jumpMenu('parent',this,0)">
<?php
    if(($_GET['bhs'] == "indonesian") || (empty($_GET['bhs']))) {
?>
        <option value="?bhs=indonesian" selected>Indonesian</option>
        <option value="?bhs=djawa">Djawa</option>
<?php
        } else {
?>
        <option value="?bhs=indonesian">Indonesian</option>
        <option value="?bhs=djawa" selected>Djawa</option>
<?php
        } // end else
?>
    </select>
</form>
<p>
<?php
    if($_GET['bhs'] == "djawa") {
        $a = include "language/djawa.php";
    } else if ($_GET['bhs'] == "indonesian") {
        $a = include "language/indonesian.php";
    } else {
        $a = include "language/indonesian.php";
    }
}
```

```

echo "$language $welcome<br>$question<br>$home";
?>
</BODY>
</HTML>

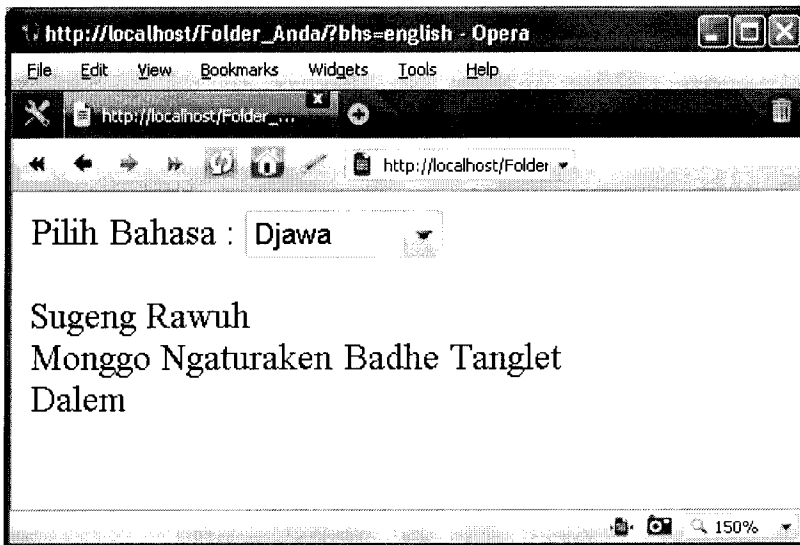
```

djawa.php

```

<?php
$welcome = "Sugeng Rawuh";
$question = "Monggo Ngaturaken Badhe Tanglet";
$home = "Dalem";
?>

```



Gambar 7.1 Bahasa Jawa

Indonesia.php

```

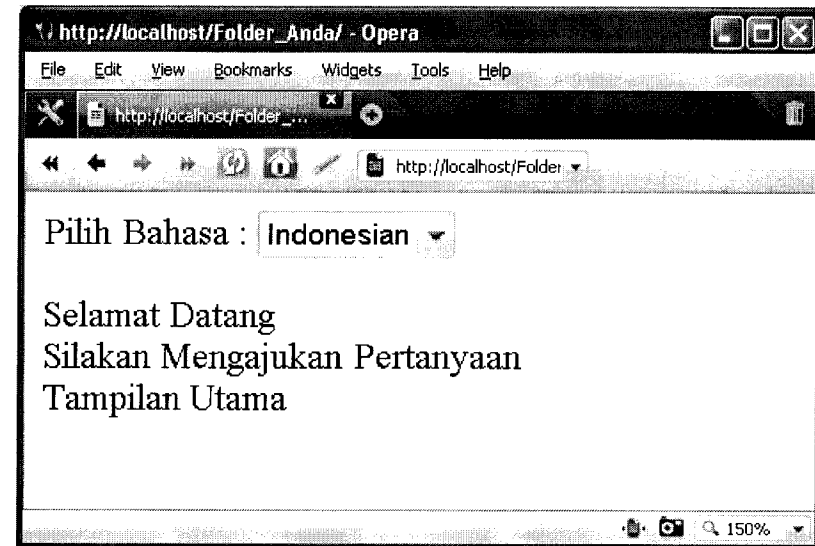
<?php
$welcome = "Selamat Datang";
$question = "Silakan Mengajukan Pertanyaan";

```

```

$home = "Tampilan Utama";
?>

```



Gambar 7.2 Bahasa Indonesia

7.2 CAPTCHA SECURITY IMAGES

Pendaftaran member pada e-mail, kegiatan download file pada suatu situs, atau aktivitas lain yang berhubungan dengan jaringan internet pada umumnya akan berhadapan dengan sistem keamanan untuk memastikan bahwa pengunjung yang sedang browsing pada situs tersebut tidak menyebarkan spam, virus, atau program yang dapat bersifat merusak. Oleh karenanya, Anda akan belajar untuk membuat sebuah Images Security. Pada PHP, disebut dengan **Captcha Security Images**. Captcha Security Images hanya dapat berjalan pada aplikasi PHP versi 5 ke atas.

Contoh program:

Input.htm

```

<HTML>
<HEAD>

```

```

<TITLE>Penerapan Aplikasi</TITLE>
</HEAD>
<BODY>
<form action="SendImages.php" method="post">
<br />
Kode Keamanan: <input id="security_code" name="security_code"
type="text" /><br />
<input type="submit" name="submit" value="Masuk" />
</form>
</BODY>
</HTML>

```

CaptchaSecurityImages.php

```

<?php
session_start();

class CaptchaSecurityImages {
var $font = 'jigsawtr.ttf';

function generateCode($characters) {
/* Mendaftar semua karakter yang mungkin, huruf hidup, vokal dan karakter
yang yang serupa telah dipindahkan*/
$possible = '23456789bcdfghjkmnpqrstvwxyz';
$code = "";
$i = 0;
while ($i < $characters) {
$code .= substr($possible, mt_rand(0, strlen($possible)-1), 1);
$i++;
}
return $code;
}

```

```

}

function CaptchaSecurityImages($width='120',$height='40',$characters='6') {
$code = $this->generateCode($characters);

/* Ukuran huruf akan menjadi 75 % sesuai dengan tinggi gambar */
$font_size = $height * 0.75;

$image = @imagecreate($width, $height) or die('Cannot Initialize new GD
image stream');

/* Mengatur pewarnaan */
$background_color = imagecolorallocate($image, 255, 255, 255);
$text_color = imagecolorallocate($image, 20, 40, 100);
$noise_color = imagecolorallocate($image, 100, 120, 180);

/* Menciptakan background secara acak */
for( $i=0; $i<($width*$height)/3; $i++ ) {
imagefilledellipse($image, mt_rand(0,$width), mt_rand(0,$height), 1, 1,
$noise_color);
}

/* Menciptakan garis pada background secara acak */
for( $i=0; $i<($width*$height)/150; $i++ ) {
imageline($image, mt_rand(0,$width), mt_rand(0,$height), mt_rand
(0,$width), mt_rand(0,$height), $noise_color);
}

/* Menciptakan textbox dan menambahkan teks */
$textbox = imageftbbox($font_size, 0, $this->font, $code);
$x = ($width-$textbox[4])/2;
$y = ($height-$textbox[5])/2;
imagefttext($image, $font_size, 0, $x, $y, $text_color, $this->font, $code);

/* Keluaran captcha images ke dalam browser */
imagejpeg($image);
imagedestroy($image);
}

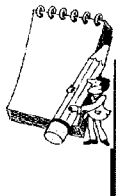
```

```

$_SESSION['security_code'] = $code;
}
}

$width = isset($_GET['width']) ? $_GET['width'] : '120';
$height = isset($_GET['height']) ? $_GET['height'] : '40';
$characters = isset($_GET['characters']) ? $_GET['characters'] : '7';
header('Content-Type: image/jpeg');
$captcha = new captchaSecurityImages($width,$height,$characters);
?>

```



Pilih font yang mudah dibaca dan letakkan font dalam direktori yang sama pada sintaks yang disimpan.

SendImages.php

```

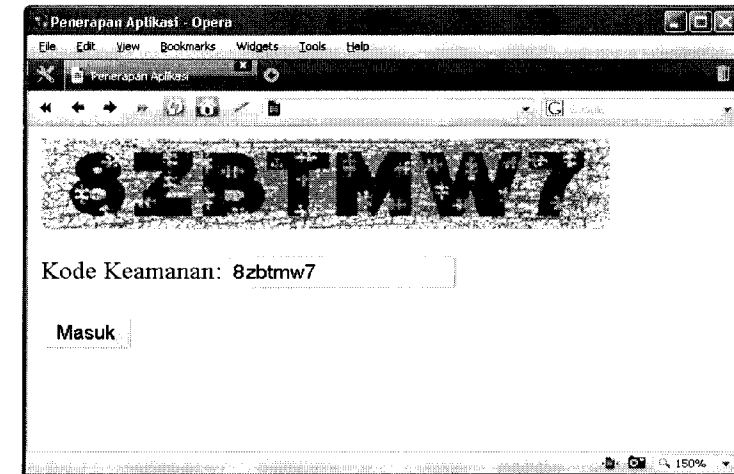
<?php
session_start();
if( isset($_POST['submit'])) {
    if(($_SESSION['security_code'] == $_POST['security_code']) &&
(!empty($_SESSION['security_code'])) ) {
        // Masukan kode yang Anda masukkan dari form ini
        echo 'Kode keamanan yang Anda masukkan benar';
    } else {
        // Pesan error jika masukan kode yang Anda masukkan salah
        echo 'Maaf kode yang Anda masukkan salah<br />';
        include "input.htm";
    }
}

```

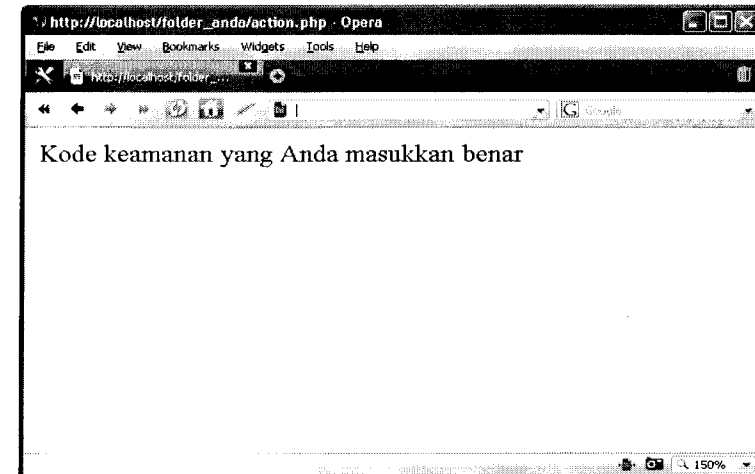
```

} else {
    include "input.htm";
}
?>

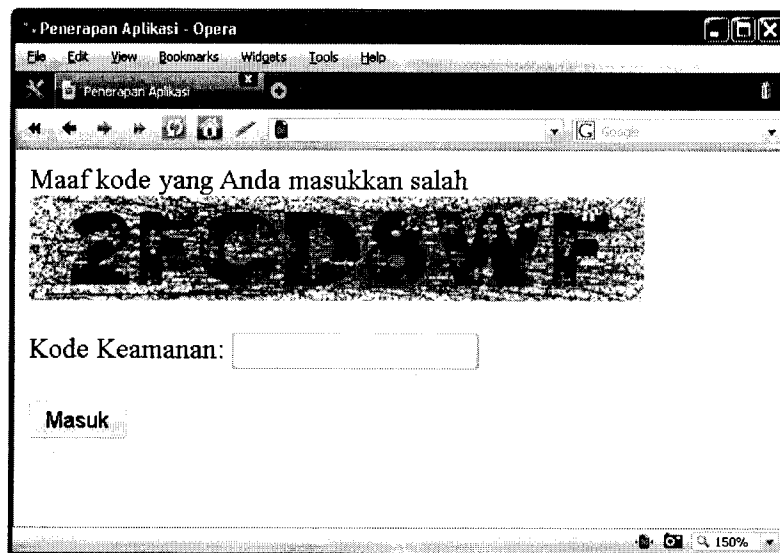
```



Gambar 7.3 Tampilan Form Kode Keamanan



Gambar 7.4 Tampilan Konfirmasi Kode Keamanan Jika Benar



Gambar 7.5 Tampilan Konfirmasi Kode Keamanan Jika Salah

Pengisian kode keamanan yang salah pada fitur Captcha Security Images akan memunculkan kode keamanan yang baru untuk diisi kembali.

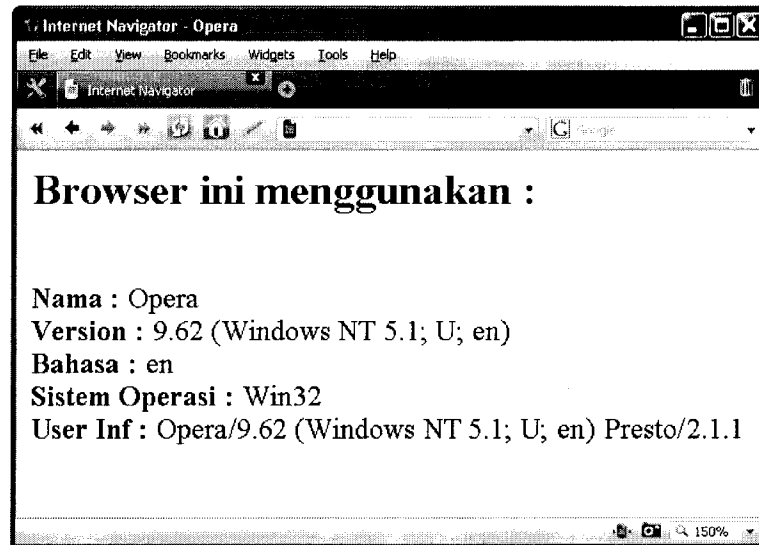
```
else {  
    // Pesan error jika masukan kode yang Anda masukkan salah  
    echo 'Maaf kode yang Anda masukkan salah<br />';  
    include "input.htm";  
}
```

7.3 NAVIGATOR INFORMATION

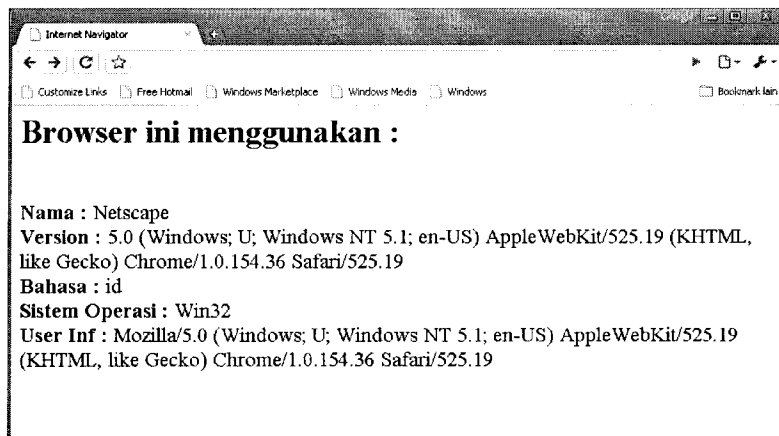
Untuk mengetahui spesifikasi browser yang digunakan, Anda dapat menggunakan *javascript* berikut ini. Sebagai contoh, pada penggunaan browser Opera, Google Chrome, Mozzila, dan Internet Explore 7.

```
<HTML>
<HEAD>
<TITLE>Internet Navigator</TITLE>
</HEAD>
```

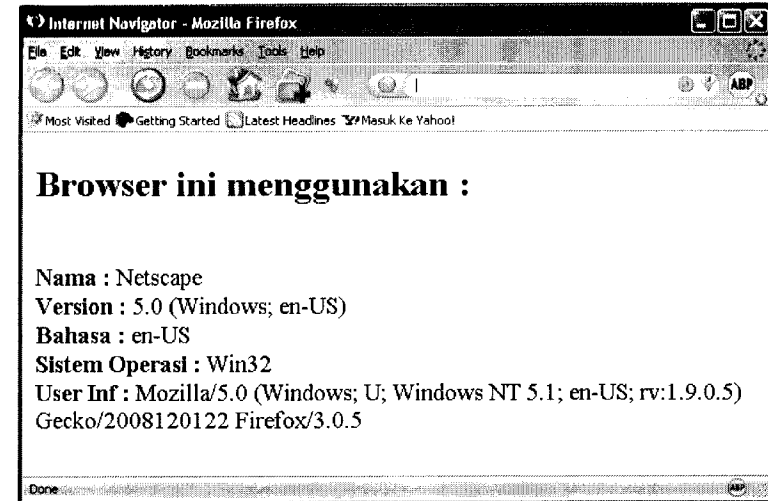
```
<BODY>
<SCRIPT LANGUAGE = "JavaScript">
<!--
    document.write("<H2>Browser ini menggunakan : </H2><BR>");
    document.write("<B>Nama : </B>" + navigator.appName +
        "<BR>");
    document.write("<B>Version : </B>" + navigator.appVersion +
        "<BR>");
    document.write("<B>Bahasa : </B>" + navigator.language +
        "<BR>");
    document.write("<B>Sistem Operasi : </B>" + navigator.platform +
        "<BR>");
    document.write("<B>User Inf : </B>" + navigator.userAgent +
        "<BR>");
//-->
</SCRIPT>
</BODY>
</HTML>
```



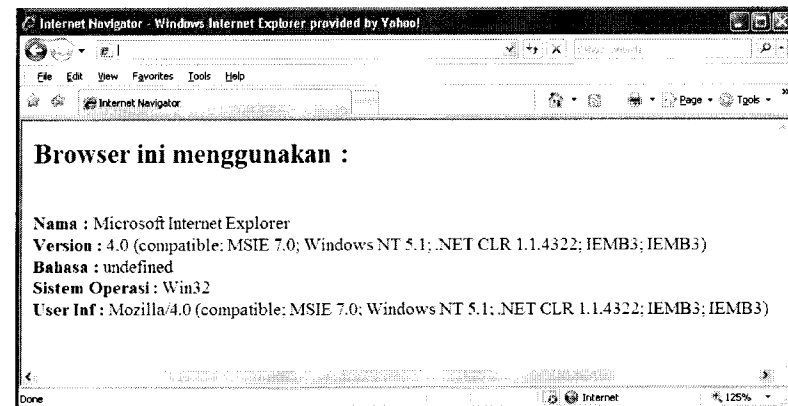
Gambar 7.6 Opera Browser Information



Gambar 7.7 Google Chrome Browser Information



Gambar 7.8 Mozilla Firefox Browser Information



Gambar 7.9 Internet Explorer Browser Information

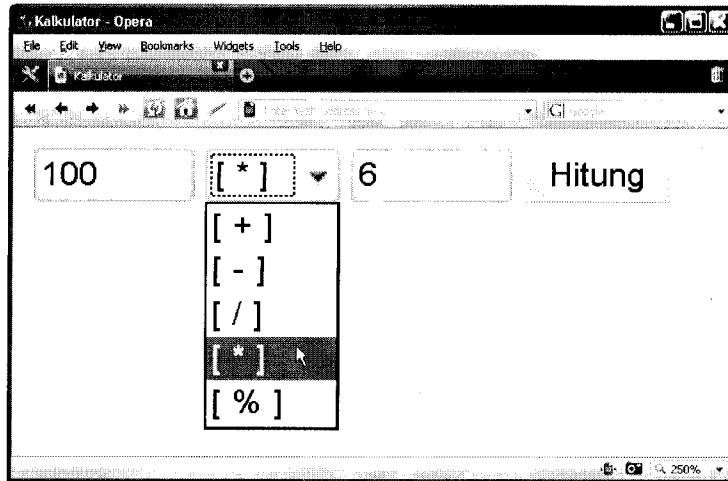
7.4 KALKULATOR DENGAN PHP

Tidak hanya Pascal, VB, dan bahasa pemrograman baik terstruktur dan objek lainnya yang bisa membuat fungsi penghitungan seperti kalkulator. PHP juga dapat melakukan penghitungan seperti pada mesin kalkulator.

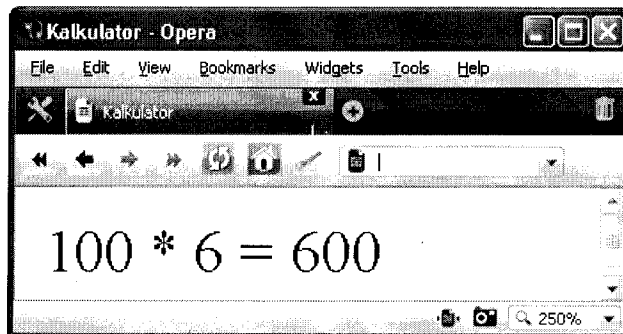
Berikut sintaks program PHP tersebut:

```
<HTML>
<HEAD>
<TITLE>Kalkulator</TITLE>
</HEAD>
<BODY>
<?PHP
if(!$_POST['act'])
{
echo '
<form method="post" action="'. $_SERVER['PHP_SELF']. ">
<input type="text" name="angka1" size="7">
<select name="act">
<option value="add">[ + ]</option>
<option value="sub">[ - ]</option>
<option value="div">[ / ]</option>
<option value="mul">[ * ]</option>
<option value="pro">[ % ]</option>
</select>
<input type="text" name="angka2" size="7">
<input type="submit" name="doit" value="Hitung">
</form>
';
}else{
if($_POST['act'] == 'add')
{
$calc = ($_POST['angka1']+$_POST['angka2']);
$op = '+';
```

```
}
if($_POST['act'] == 'sub')
{
$calc = ($_POST['angka1']-$_POST['angka2']);
$op = '-';
}
if($_POST['act'] == 'div')
{
$calc = ($_POST['angka1']/$_POST['angka2']);
$op = '/';
}
if($_POST['act'] == 'mul')
{
$calc = ($_POST['angka1']*$_POST['angka2']);
$op = '*';
}
if($_POST['act'] == 'pro')
{
$calc = (($_POST['angka2']*$_POST['angka1'])/100);
$op = '% of';
}
echo $_POST['angka1'].' '.$op.' '.$_POST['angka2'].' = '.$calc;
}
?>
</BODY>
</HTML>
```



Gambar 7.10 Form Kalkulator dengan \$calc



Gambar 7.11 Hasil Penghitungan

7.5 MEMBUAT KALENDER

Anda juga dapat mencari jumlah hari dalam bulan tertentu dengan fungsi khusus yang disediakan oleh PHP. Function yang digunakan adalah **cal_days_in_month()**;

Function ini memiliki sintaks sebagai berikut:

```
cal_days_in_month(a, b, c);
```

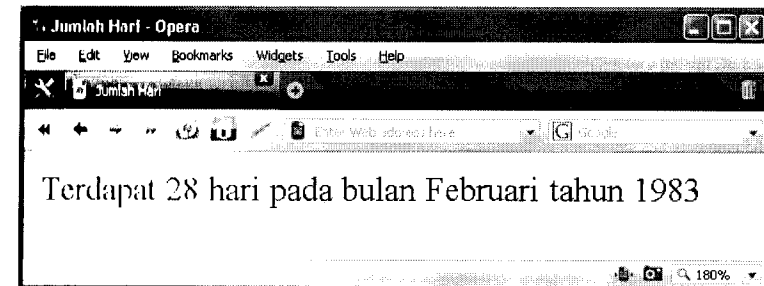
Keterangan:

- a adalah jenis kalender yang diinginkan.
- b adalah bilangan yang menunjukkan jenis bulan.
- c adalah tahun.

Untuk parameter a, PHP menyediakan 4 jenis kalender yang dapat digunakan, yaitu: kalender Perancis (**CAL_FRENCH**), kalender Gregorius (**CAL_GREGORIAN**), kalender Yahudi (**CAL_JEWISH**), dan kalender Julian (**CAL_JULIAN**). Kalender yang sering dipakai di dunia adalah kalender Gregorian.

Sebagai contoh, Anda akan mencari jumlah hari pada bulan Februari tahun 1983, ketikkan sintaks program sebagai berikut:

```
<HTML>
<HEAD>
<TITLE>Jumlah Hari</TITLE>
</HEAD>
<BODY>
<?php
$jumHari = cal_days_in_month(CAL_GREGORIAN, 2, 1983);
echo "Terdapat ".$jumHari." hari pada bulan Februari tahun 1983";
?>
</BODY>
</HTML>
```



Gambar 7.12 Jumlah Hari pada Februari 1983

Anda juga dapat menampilkan kalender pada browser dengan mengetikkan programnya berikut ini:

```
<HTML>
<HEAD>
<TITLE>Kalender</TITLE>
</HEAD>
<BODY>
<?php
//bulan ini
$month=date("m");
//Tahun ini
$year=date("Y");
//hari ini
$day=date("d");
//cek jumlah hari tahun ini
$endDate=date("t",mktime(0,0,0,$month,$day,$year));
//style untuk table
echo '
<style>
td {
font-size:11;
font-family:arial;
}
</style>
';
//tabel yang digunakan untuk menulis tanggal hari ini
echo '<table align="center" border="0" width="100%" cellpadding=2
cellspacing=1 style=""><tr><td align=left>';
```

```
echo date("D, d M Y ",mktime(0,0,0,$month,$day,$year));
echo '</td></tr></table>';

//tabel yang digunakan untuk menulis hari
echo '
<table align="left" border="0" width="20%" cellpadding=3 cellspacing=2
style="border:2px solid grey">
<tr bgcolor="yellow">
<td align=center><font color=red>Minggu</font></td>
<td align=center>Senin</td>
<td align=center>Selasa</td>
<td align=center>Rabu</td>
<td align=center>Kamis</td>
<td align=center><font color=blue>Jumat</font></td>
<td align=center>Sabtu</td>
</tr>
';
// mengecek tanggal 1 bulan ini terdapat pada hari ke berapa setelah itu
tambahkan cell td sebanyak var $s
$s=date ("w", mktime (0,0,0,$month,1,$year));
for ($ds=1;$ds<=$s;$ds++) {
echo "<td style=\"font-family:Calibri;color:red\" width=\"15%\"
align=center valign=middle bgcolor=\"#FFFFFF\">
</td>";
}
//penulisan tanggal
for ($d=1;$d<=$endDate;$d++) {
//jika nilai $d (tanggal) adalah hari minggu (0) dibuat baris baru <tr>
if (date("w",mktime (0,0,0,$month,$d,$year)) == 0) { echo "<tr>"; }
```

```
//default warna font
$fontColor="black";

//jika tanggal menyatakan hari minggu warna huruf merah
if (date("D",mktime (0,0,0,$month,$d,$year)) == "Sun")
{ $fontColor="red"; }

//jika tanggal menyatakan hari jum'at warna huruf biru
if (date("D",mktime (0,0,0,$month,$d,$year)) == "Fri")
{ $fontColor="blue"; }

//menulis tanggal
echo "<td style=\"font-family:arial;color:blue\" width=\"15%\"
align=center valign=middle> <span
style=\"color:$fontColor\">$d</span></td>";

//jika tanggal menyatakan hari sabtu (6) akhiri baris </tr>
if (date("w",mktime (0,0,0,$month,$d,$year)) == 6) { echo "</tr>"; }
}

//akhir pembuatan tabel
echo '</table>';

?>

</BODY>
</HTML>
```



Gambar 7.13 Tampilan Kalender dengan PHP

7.6 ORGANISASI FILE

Selain dengan `file.txt`, PHP ternyata juga dapat dihubungkan dengan file `.rtf` dan beberapa jenis file lainnya. Sebelum memulai membuat skrip HTML dan PHP, Anda harus membuat dokumen terlebih dahulu dengan Wordpad atau Microsoft Office, lalu menyimpannya dengan nama (misalnya) `info.rtf`.

Berikut sintaks program `index.html`:

```
<HTML>
<HEAD>
<TITLE>FILE TEXT</TITLE>
</HEAD>
<BODY>
<h2>Pendaftaran Data Mahasiswa</h2>
<form method="post" action="kirim.php">
```

```

<table>
<tr><td>NAMA MAHASISWA</td><td><input type="text"
name="nama"></td></tr>
<tr><td>NIM</td><td><input type="text" name="nimmhs"></td></tr>
<tr><td>FAKULTAS</td><td><input
name="fakultas"></td></tr>
<tr><td>JURUSAN</td><td><input
name="jurusan"></td></tr>
<tr><td>UNIVERSITAS</td><td><input
name="universitas"></td></tr>
<tr><td>IP          SEMESTER</td><td><input
name="ip"></td></tr>
<tr><td>PERINGKAT</td><td><input
name="peringkat"></td></tr>
<tr><td>POTONGAN      SPP</td><td><input
name="spp"></td></tr>
<tr><td></td><td><input type="submit" name="submit" value="Input
Data"></td></tr>
</table>
</form>
</BODY>
</HTML>

```

Sintaks program kirim.php:

```

<?php
$nama = $_POST['nama'];
$nimmhs = $_POST['nimmhs'];
$fakultas = $_POST['fakultas'];
$jurusan = $_POST['jurusan'];
$universitas = $_POST['universitas'];
$ip = $_POST['ip'];

```

```

$peringkat = $_POST['peringkat'];
$spp = $_POST['spp'];

$document = file_get_contents("info.rtf");

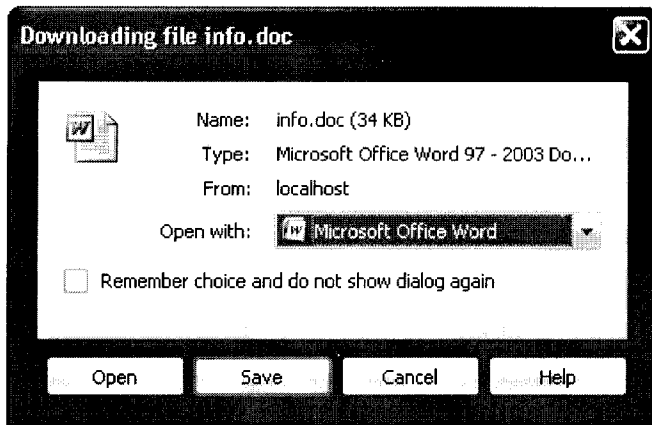
$document = str_replace("%%NAMA%%", $nama, $document);
$document = str_replace("%%NIMMHS%%", $nimmhs, $document);
$document = str_replace("%%JURUSAN%%", $fakultas, $document);
$document = str_replace("%%FAKULTAS%%", $jurusan, $document);
$document = str_replace("%%IP%%", $ip, $document);
$document = str_replace("%%PERINGKAT%%", $peringkat, $document);
$document = str_replace("%%SPP%%", $spp, $document);

header("Content-type: application/msword");
header("Content-disposition: inline; filename=info.rtf");
header("Content-length: " . strlen($document));
echo $document;
?>

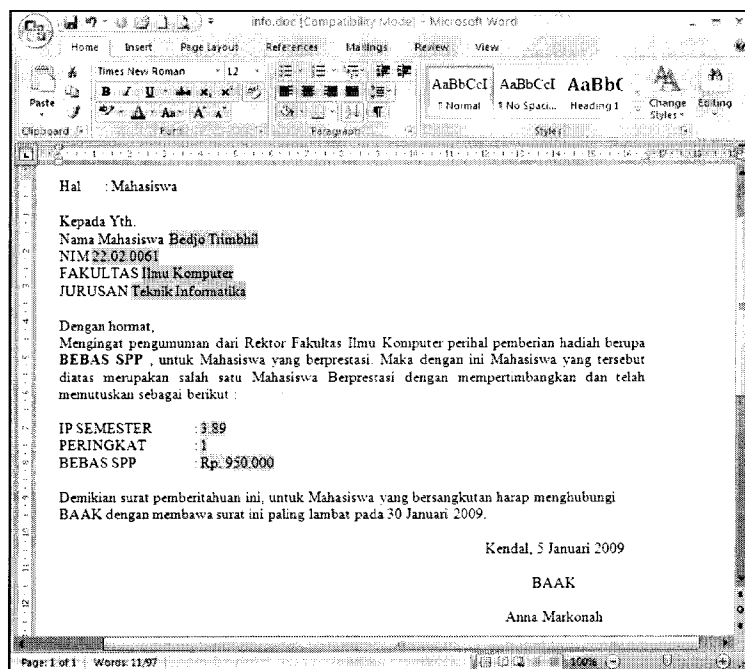
```

The screenshot shows a web browser window with a form titled "Pendaftaran Data Mahasiswa". The form has several input fields, each with a label and a value. The labels are: NAMA MAHASISWA, NIM, FAKULTAS, JURUSAN, UNIVERSITAS, IP SEMESTER, PERINGKAT, and POTONGAN SPP. The values are: Bedjo Trimbhil, 22.02.0061, Ilmu Komputer, Teknik Informatika, UNIMIK, 3.89, 1, and 950.000. Below the form is a button labeled "Input Data".

Gambar 7.14 Form Input Data



Gambar 7.15 Kotak Pesan Download File



Gambar 7.16 Hasil Cetak Input Data pada Microsoft Word

7.7 FILTER IMAGES DENGAN PHP

Dalam perkembangannya, PHP tidak hanya menyerukan segi keamanan dan fungsi OOP kepada penggemarnya. PHP juga dapat digunakan untuk mengolah gambar, meskipun masih sebatas mengubah tampilan gambar. Seperti halnya fasilitas Image Adjustment pada Adobe Photoshop, PHP memiliki fungsi untuk kebutuhan yang sama dengan menggunakan **IMAGE_FILTER_**. Sebelum menggunakan fasilitas IMAGE FILTER pada PHP, Anda disarankan agar menyiapkan gambar terlebih dahulu dengan format **JPEG, BMP, PNG**, atau lainnya.

Beberapa fungsi IMAGE FILTER yang tersedia di PHP, antara lain:

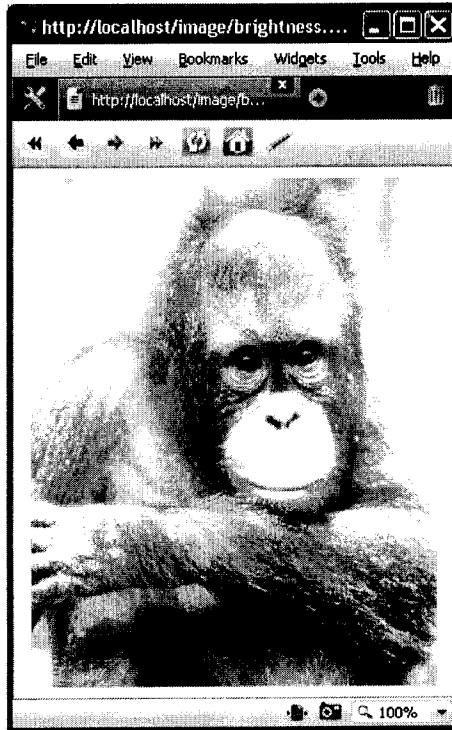
- **IMG_FILTER_EDGEDETECT**
- **IMG_FILTER_BRIGHTNESS**
- **IMG_FILTER_COLORIZE**
- **IMG_FILTER_EMOSS**
- **IMG_FILTER_GAUSSIAN_BLUR**
- **IMG_FILTER_GRAYSCALE**
- **IMG_FILTER_NEGATE**
- **IMG_FILTER_MEAN_REMOVAL**
- **IMG_FILTER_SMOOTH**

Contoh program dengan fungsi Brightness:

```
<?php
// mengatur tingkat brightness pada gambar
header("Content-type: image/jpeg");

$image = imagecreatefromjpeg('orangutan.jpg');
imagefilter($image, IMG_FILTER_BRIGHTNESS, 70);
imagejpeg($image);

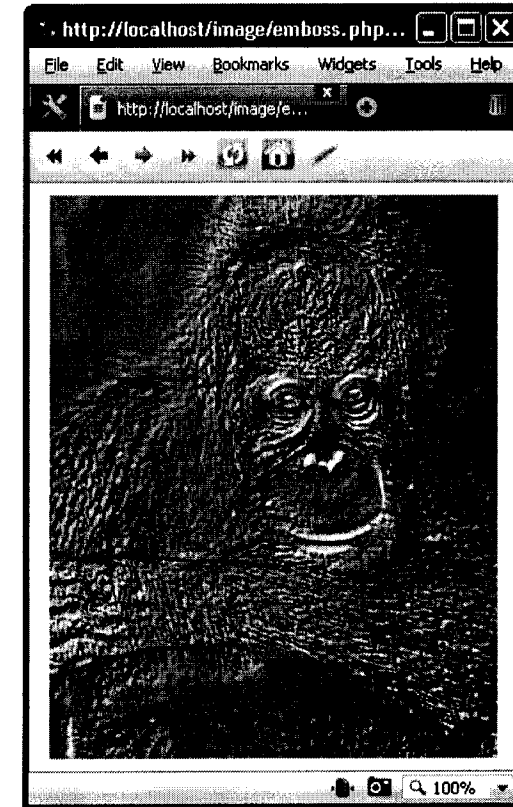
?>
```



Gambar 7.17 Pengolahan Gambar dengan IMAGE_FILTER_BRIGHTNESS

Contoh program dengan fungsi Emboss:

```
<?php
// image dengan efek emboss
header("Content-type: image/jpeg");
$image = imagecreatefromjpeg('orangutan.jpg');
imagefilter($image, IMG_FILTER_EMOSS);
imagejpeg($image);
?>
```



Gambar 7.18 Pengolahan Gambar dengan IMAGE_FILTER_EMOSS

Contoh program dengan fungsi Negatif:

```
<?php
// membuat image negatif
header("Content-type: image/jpeg");
$image = imagecreatefromjpeg('orangutan.jpg');
imagefilter($image, IMG_FILTER_NEGATE);
imagejpeg($image);
?>
```



Gambar 7.19 Pengolahan Gambar dengan IMAGE_FILTER_NEGATE

Penutup

Sampai pada bagian ini, Anda telah belajar banyak mengenai konsep dan aplikasi web dengan PHP programming. Sejak awal, Anda telah berkenalan dengan PHP, konfigurasi, dan elemen pada PHP. Selain itu, Anda juga telah memahami konstanta, operator, ekspresi, hingga konsep class. Pada bagian sebelumnya juga telah diberikan contoh penerapan PHP pada aplikasi web. Dari contoh aplikasi yang telah diberikan tersebut, Anda diharapkan dapat mengembangkan kemampuan web programming dengan PHP.

Kami berharap agar Anda dapat memahami materi yang diberikan dengan mudah. Jika menemui kendala dalam memahami materi yang disampaikan, Anda dapat menyampaikannya melalui alamat penulis. Anda juga dapat mengikuti mailing list, forum yang ada di internet, atau berkunjung ke alamat: <http://php.net> untuk mendapatkan referensi lengkap dan mengembangkan kemampuan PHP programming Anda. Selamat belajar!